Advant® OCS
with Master software

AMPL Control Configuration Commands for Structure Builder

Version 2.7

User’s Guide

3BSE 013 931R301

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant, AdvaBuild, and AdvaInform are registered trademarks of ABB Asea Brown Boveri Ltd., Switzerland.

Advant Controller, Advant Fieldbus, Advant Station, AdvaBuild On-line Builder, AdvaCommand, and AdvaSoft are trademarks of ABB Asea Brown Boveri Ltd., Switzerland.

ABB Master, MasterAid, MasterBus, MasterPiece, and MasterView are registered trademarks of ABB Asea Brown Boveri Ltd., Switzerland.

MasterGate, MasterFieldbus, and MasterNet are trademarks of ABB Asea Brown Boweri Ltd., Switzerland.

Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.

Windows, Windows for Workgroups, and Windows NT are trademarks of Microsoft Corporation.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Automation Products GmbH. ABB Automation Products GmbH assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Automation Products GmbH be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Automation Products GmbH be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Automation Products GmbH’s written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Copyright © ABB Automation Products GmbH 1999

3BSE001264/E

Template: 3BSE001286/F

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 Before You Begin 1-1
1.2 Equipment Requirements 1-3
1.3 How to Use This Book 1-3
1.4 Conventions 1-4
1.5 Related Documentation 1-5
1.6 Release History 1-6
1.7 Terminology 1-7
1.8 Product Overview 1-8
1.9 User Interface 1-9

Chapter 2 - Installation

Chapter 3 - Configuration/Application Building

3.1 Considerations 3-1
  3.1.1 About the Advant Object Technology 3-1
  3.1.2 Aspects for OCS Control Configuration 3-1
  3.1.3 Parameter References 3-2
  3.1.4 DB Element Handling 3-3
  3.1.5 Circuit Handling 3-4
3.2 Capacity and Performance 3-4
3.3 Application Start-up 3-5

Chapter 4 - Runtime Operation

4.1 Operating Overview 4-1
  4.1.1 Create (New) an Advant Object 4-1
  4.1.2 Open an Advant Object 4-2
  4.1.3 Delete an Advant Object 4-3
  4.1.4 Copy an Advant Object 4-3
  4.1.5 Buttons and Popup Menu in Object Window 4-4
  4.1.6 New Aspect 4-5
  4.1.7 Open Aspect 4-5
  4.1.8 Print Aspect 4-5
  4.1.9 Delete Aspect 4-5
  4.1.10 Additional Commands 4-5
4.2 Runtime Tutorial 4-6
4.3 Operating Instructions 4-6
  4.3.1 Set Options, Project OCS Defaults Aspect 4-6

CONTENTS (continued)

4.3.2 Create Project OCS Defaults Aspect 4-9
4.3.3 Attach Project OCS Defaults Aspect 4-11
4.3.4 Open Project OCS Defaults Aspect 4-11
4.3.5 Delete Project OCS Defaults Aspect 4-13
4.3.6 Create Aspect 4-13
    4.3.6.1 Type Circuit Reference-Handling 4-15
4.3.7 Attach Aspect 4-15
4.3.8 Copy Aspect 4-16
4.3.9 Open Aspect 4-18
4.3.10 Print Aspect 4-21
4.3.11 Delete Aspect 4-22
4.3.12 Create Node Control Software Aspect 4-23
4.3.13 Attach Node Control Software Aspect 4-25
4.3.14 Copy Node Control Software Aspect 4-25
4.3.15 Open Node Control Software Aspect 4-26
4.3.16 Print Node Control Software Aspect 4-28
4.3.17 Delete Node Control Software Aspect 4-29
4.3.18 Convert Node Control Software Aspect 4-30
4.3.19 Extract Node Control Software Aspect 4-30
4.3.20 Split Node Control Software Aspect 4-31
4.3.21 Concatenate 4-32
4.3.22 Lock/Unlock 4-32
4.3.23 Configuring Advant Objects for Library 4-33
    4.3.23.1 Objects for Insertion of Process Control Functions via Drag and Drop 4-33
    4.3.23.2 Objects for Insertion of DB Elements via Drag and Drop 4-34
    4.3.23.3 Objects for Insertion of DB Elements via Generate DB 4-34
    4.3.23.4 PCPageBreak Parameter Aspect 4-34
    4.3.23.5 Instance Names With/Without Absolute Reference Designation 4-35
4.3.24 Drag and Drop (Insert) 4-35
4.3.25 Locate Circuit 4-39
4.3.26 Circuit Usage List 4-41
4.3.27 Parameter References 4-42
    4.3.27.1 Configure Parameter References 4-42
    4.3.27.2 Update Parameter 4-45
    4.3.27.3 Generate DB 4-47
    4.3.27.4 Drag and Drop / Locate Circuit 4-48

CONTENTS (continued)

4.3.28 DB Element Handling 4-49

4.3.28.1 Configure DB Element Aspect for Board Advant Object 4-49

4.3.28.2 Configure DB Element Aspect for Signal Advant Object 4-51

4.3.28.3 Configure DB Element Aspect for Other Kind of DB Element 4-52

4.3.28.4 Building up OCS Structure 4-53

4.3.28.5 Generate DB 4-54

4.3.28.6 Update Parameter 4-56

4.3.28.7 Initialize 4-56

4.3.29 Execution of Commands by the Function Chart Builder 4-56

4.3.30 Export Process 4-57

4.3.31 Import Process 4-57

4.3.31.1 Create Objects and Aspects 4-57

4.3.31.2 Create Objects and Aspects From Exported Data 4-59

4.4 Runtime Operation Menus 4-59

4.4.1 Command Buttons 4-59

4.4.1.1 New… 4-59

4.4.1.2 Close 4-60

4.4.1.3 Delete 4-60

4.4.1.4 Edit… 4-60

4.4.1.5 More… 4-60

4.4.1.6 Open 4-60

4.4.2 Popup Menu Commands 4-60

4.4.2.1 Attach 4-60

4.4.2.2 Convert 4-61

4.4.2.3 Extract 4-61

4.4.2.4 Generate DB 4-61

4.4.2.5 Open 4-61

4.4.2.6 Print 4-63

4.4.2.7 Set Options 4-63

4.4.2.8 Split 4-63

4.4.2.9 Update Parameter 4-63

4.4.2.10 Initialize 4-64

4.4.2.11 Concatenate 4-64

4.4.2.12 Lock/Unlock 4-64

4.4.2.13 Locate Circuit 4-64

4.4.2.14 Update FCB Data 4-65

CONTENTS (continued)

Chapter 5 - Maintenance

5.1 Error Messages 5-1
5.2 Log-File for Update Parameter/Generate DB/Initialize 5-2
5.3 Important Environment Variables 5-6
  5.3.1 Environment Variable TC$LEV1 5-6
  5.3.2 Environment Variable ADVABLDENV 5-6
  5.3.3 Environment Variables TMP and TEMP 5-6

Appendix A - References

A.1 Aspect Types and Integrated Functions A-1
A.2 Commands and Views of Aspect Types A-4

INDEX

ILLUSTRATIONS

Figure 1-1.AMPL Control Configuration Functions 1-2
Figure 1-2.Organization of Manual 1-3
Figure 4-1.Project Pool Window 4-2
Figure 4-2.Object Window 4-3
Figure 4-3.Object Window with Command Popup Menu 4-4
Figure 4-4.New Aspect Dialog 4-5
Figure 4-5.Set Options Dialog 4-6
Figure 4-6.New Project Dialog 4-10
Figure 4-7.Object Window Showing Project OCS Defaults Aspect 4-10
Figure 4-8.Copy Dialog for Attach Project OCS Default Aspect 4-11
Figure 4-9.Open View Summary Form on Project OCS Defaults Aspect 4-12
Figure 4-10.Open View Notes Text on Project OCS Defaults Aspect 4-12
Figure 4-11.Initial Object Window 4-13
Figure 4-12.New Circuit Aspect 4-14
Figure 4-13.New Circuit Dialog 4-14
Figure 4-14.Object Window with Added Circuit Aspect 4-15
Figure 4-15.Copy Dialog for Attach Circuit Aspect 4-16
Figure 4-16.Paste Special Dialog for Circuit Aspect 4-17
Figure 4-17.Copy Dialog for Circuit Aspect 4-17
Figure 4-18.Open View Function Chart on Circuit Aspect 4-18

ILLUSTRATIONS (continued)

Figure 4-19.Open View Summary Form for Circuit Aspect 4-19
Figure 4-20.Open View Node Function Chart for Circuit Aspect 4-20
Figure 4-21.Open View Source Text on Circuit Aspect/Extended View Dialog 4-20
Figure 4-22.Open View Source Text on Circuit Aspect / AS100-EDIT 4-21
Figure 4-23.Print Circuit Aspect 4-22
Figure 4-24.Delete Circuit Aspect 4-23
Figure 4-25.New Node Control Software Aspect 4-23
Figure 4-26.Add Node Control Software Aspect / New Node Dialog 4-24
Figure 4-27.Object Window with Added Node Control Software Aspect 4-24
Figure 4-28.Copy Dialog for Attach Node Control Software Aspect.............................. 4-25
Figure 4-29.Open Node Control Software Aspect with Node Function Chart View........ 4-26
Figure 4-30.Open Node Control Software Aspect with Summary Form View 4-27
Figure 4-31.Open Node Control Software Aspect with View Source Text / Extended View 4-28
Figure 4-32.Open Node Control Software Aspect with View Source Text / AS100-Edit 4-28
Figure 4-33.Print Node Control Software Aspect, Node Documents View 4-29
Figure 4-34.Delete Node Control Software Aspect 4-29
Figure 4-35.Convert Node Control Software Aspect... 4-30
Figure 4-36.Extract Node Control Software Aspect 4-31
Figure 4-37.Split Node Control Software Aspect 4-32
Figure 4-38.Drag and Drop 4-36
Figure 4-39.Example of OCS Structure with PC Item Designation 4-40
Figure 4-40.Circuit Usage List File CiStatus.txt Opened with AS100-Edit 4-41
Figure 4-41.Reference Configuration of TC for Circuit Aspect of a Consumer Advant Object (Example).. 4-43
Figure 4-42.Reference Configuration of Signal TC for DB Element Aspect of a Signal Advant Object (Example) 4-43
Figure 4-43.TC Symbol Update Switch Settings (Warning).. 4-45
Figure 4-44.Change Settings Information.. 4-46
Figure 4-45.Aspects of an Advant Object Representing a Board 4-49
Figure 4-46.Object Window of an Advant Object Representing a Signal 4-51
Figure 4-47.Signals, Boards, and a Group of Boards in OCS Structure.......................... 4-53
Figure 4-48.Object Window of Advant Object with DB Section Aspect 4-54
Figure 4-49.Example Import Data File for Create a Circuit Aspect 4-59

TABLES

Table 1-1. Related Documentation 1-5
Table 1-2. Release History of the AMPL Control Configuration Commands 1-6
Table 1-3. Release History of this User's Guide 1-6
Table 1-4. Functions Integrated for Different Advant Object Aspects........ 1-9
Table 3-1. Aspects for OCS Control Configuration 3-2
Table 4-1. Generated TC Names (Example)........ 4-7
Table 4-2. Generated CI Names (Example)........ 4-8
Table 4-3. Generated Node Names (Example)........ 4-8
Table 4-4. Examples of Relative Reference Designations for Locate Circuit Command . 4-40
Table 4-5. Open Command: Aspect Types and Views 4-62
Table 4-6. Print Command: Aspect Types and Views 4-63
Table A-1.Summary of Aspect Types, Integrated Functions, and Views........ A-1
Table A-2.Project OCS Defaults Aspect Type, Commands and Views A-4
Table A-3.Type Circuit Aspect Type, Commands and Views........ A-5
Table A-4.DB Element Aspect Type, Commands and Views........ A-6
Table A-5.Circuit Aspect Type, Commands and Views........ A-7
Table A-6.DB Section Aspect Type, Commands and Views A-8
Table A-7.Node Control Software Aspect Type, Commands and Views........ A-9

Chapter 1 Introduction

1.1 Before You Begin

The project engineering of process automation projects is based on the **Advant object** technology. Projects and Advant objects for control configuration are basically structured and administered by Structure Builder (STB). Control configuration data are administered by the components of AMPL Control Configuration. These components can run in two modes:

- **Independent**

  Use Application Builder as navigation tool. Work without Structure Builder and Advant Engineering Server. Creation of Advant objects is not supported in this mode. The following tools support the independent mode:

  - Application Builder
  - Function Chart Builder
  - On-line Builder
  - Bus Configuration Builder.

- **Cooperating**

  Use Structure Builder as navigation tool. Run AMPL Control Configuration commands via Structure Builder with the Advant Engineering Server. Creation of Advant objects is supported. The following tools support the cooperating mode:

  - Structure Builder
  - Application Builder
  - Function Chart Builder
  - On-line Builder
  - Bus Configuration Builder
  - Document Builder
  - Parameter Builder
  - ...

  Application Builder is not used as navigation tool, but controls cooperative tasks.

This user's guide describes the AMPL Control Configuration commands for Structure Builder, which are available in the cooperating mode.

Different other builders, such as the Parameter Builder, or collection of other builders are available to process different aspects of Advant objects.

Functions of the following components of AMPL Control Configuration are available when navigating via Structure Builder:

- Application Builder (APB)
- Function Chart Builder (FCB)
- On-line Builder (ONB)
- Bus Configuration Builder (BCB).

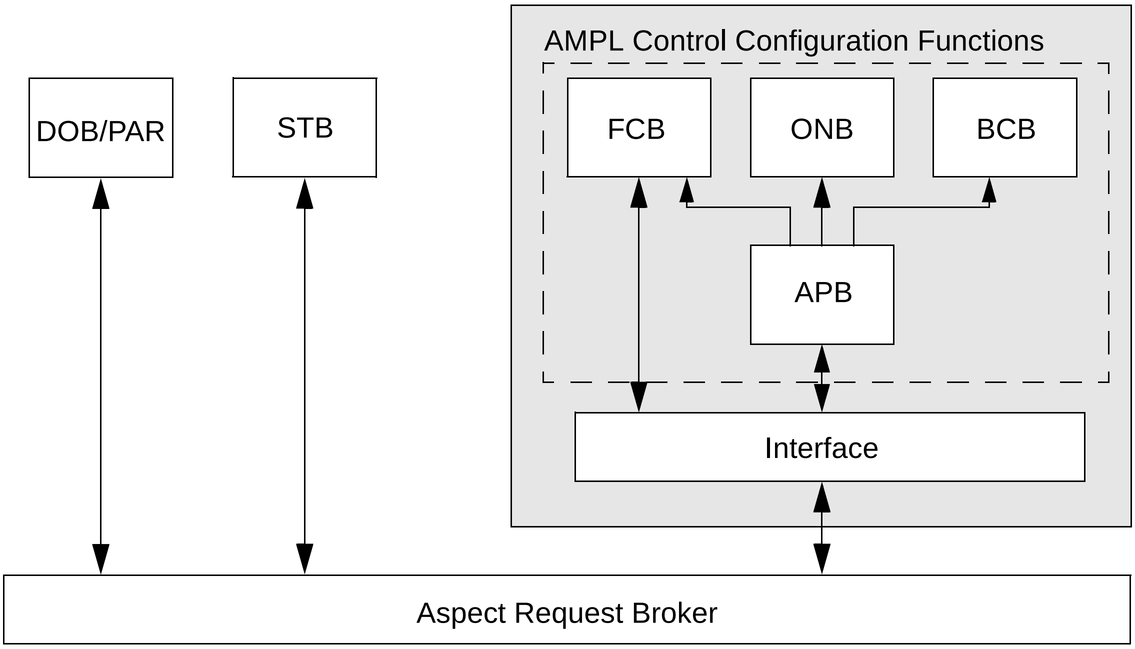


Figure 1-1. AMPL Control Configuration Functions

The control configuration functions are provided by the four builders and their integration interface. You access the control configuration functions by commands, via the Structure Builder's user interface.

The integrated functionality covers creation, deletion, modification, and documentation of control related aspects of Advant objects and administration of control configuration projects.

The guide containing tutorial, “how to” and reference information is useful both for novice and expert users.

It tells you how to use the functions available in the cooperating mode.

But it does neither tell you how to use the functions in detail nor teaches you Advant Controller or MasterPiece system programming. To learn about these topics attend system engineering courses offered by the local ABB representative and consult the manuals listed at the end of this chapter.

1.2 Equipment Requirements

See *AMPL Control Configuration Release Notes*.

1.3 How to Use This Book

This book is organized into chapters and sections as shown in Figure 1-2.

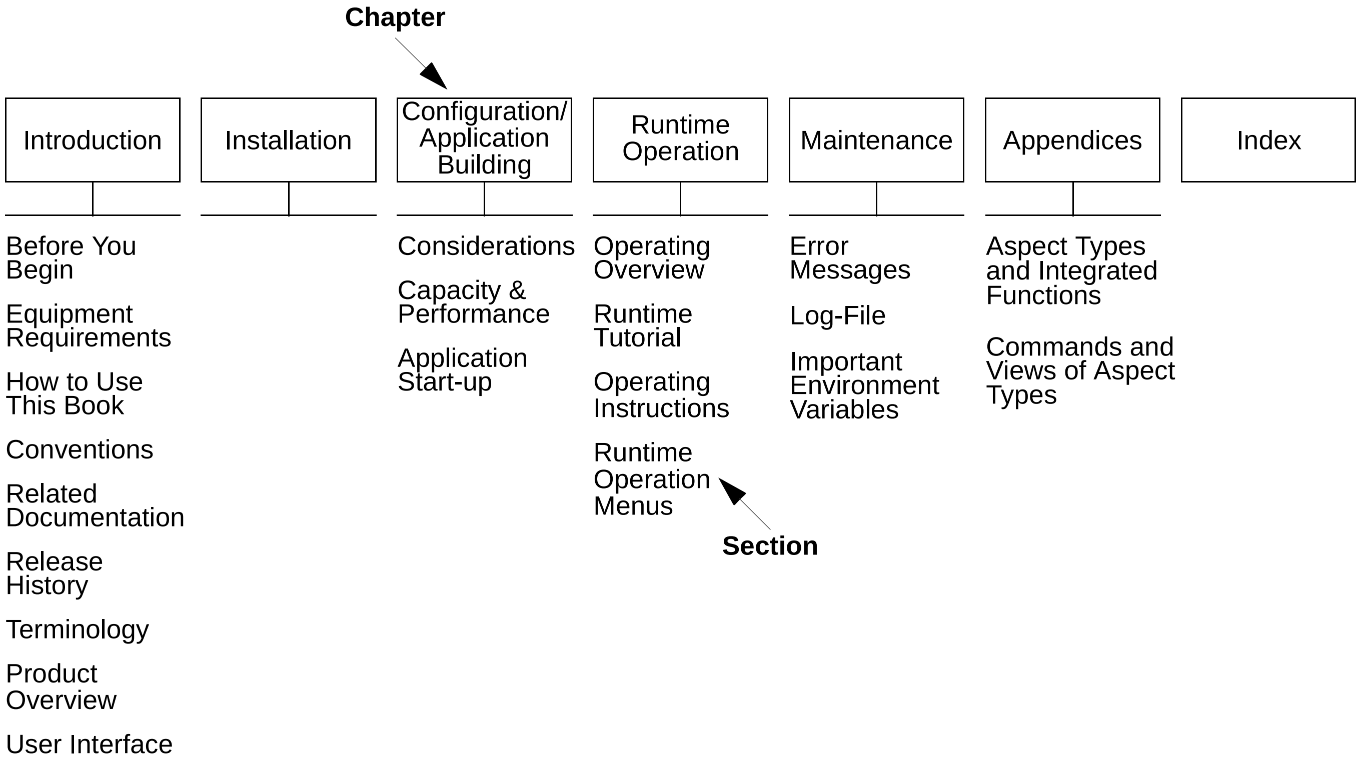


Figure 1-2. Organization of Manual

Introduction

General information about the AMPL Control Configuration commands for Structure Builder and a product overview is included in Chapter 1, Introduction.

Installation

All details concerning setup procedures you will find in Chapter 2, Installation.

Configuration and Application Building

In Chapter 3, Configuration/Application Building you find basic considerations concerning the use the functions of AMPL Configuration via Structure Builder and information about capacity, performance, and start-up of the application.

Runtime Operation

Chapter 4, Runtime Operation describes how to use the functions of AMPL Control Configuration Functions from within the Structure Builder and describes the menu commands.

Maintenance

Chapter 5, Maintenance gives you basic information on error messages that may occur during your work with AMPL Control Configuration.

Appendix A

Appendix A, References contains further reference information.

Index

The Index tells you where to find references to variety of key topics within the text.

1.4 Conventions

The names of keyboard keys, push and toggle buttons, menus, and menu items are boldfaced.

Books, manuals, and so on, referenced in the text are written in *italic*.

Substitutable arguments are written in *italic*.

Filenames, directory names, acronyms appear in all capitals.

Application windows, document windows, and dialog boxes are referenced by their names in **boldface** type, for example, **PC Section** window.

Words, which you have to type exactly in that form, are shown in **boldface** type, and if possible in CAPITAL letters, for example, **POWERSTN**.

Texts and system messages displayed on the screen are shown in `Courier` font, for example, `Select Options`.

Keyboard Interaction

When you read	You shall
• **File | Copy**	Open the **File** menu and choose the item **Copy**.
• **<Ctrl+M>**	Press the **Ctrl** key and the key labeled **M** simultaneously.

Syntax

To start a program or a command, enter the command syntax and press the **Enter** key.
Specify the directory if needed.

- Bold strings are the command names and directories; they must be typed as they appear.
- Italic strings represent substitutable arguments and directories.

1.5 Related Documentation

Table 1-1 lists all related documentation.

Table 1-1. Related Documentation

Category	Title	Description
Software	Structure Builder User's Guide	Describes the Structure Builder and how to handle Advant objects, structures, and aspects.
	Application Builder User's Guide	Describes installation of AMPL Control Configuration components; contains the operating instruction of Application Builder, Bus Configuration Builder, and support functions.
	Function Chart Builder User's Guide	Describes how to apply the Function Chart Builder for configuration and application building of Advant Controllers.
	On-line Builder User's Guide	Contains the operating instruction of On-line Builder.
	Parameter Builder User's Guide	Describes how to handle parameters for process objects and for OCS objects and how to import engineering data by the Parameter Builder.
	Bulk Data Handling and Reuse Reference Manual	Describes Bulk Data Handling functions (for example Export/Import) and Reuse
	Bus Configuration Builder User's Guide	Describes how to configure AF 100 for AdvaSoft for Windows and other Human-Machine Interfaces by the Bus Configuration Builder.
Station	Advant Station 100 Series Engineering Station User's Guide	Describes the hardware and setup of Advant Station 100 Series Engineering Station including the Advant Station 100 Series Engineering Board.
Other	Engineering Methods Reference Manual	Through an example project describes the Advant object approach, the function oriented engineering method, the work with standard solutions, and other ways of deployment of Advant Engineering Workplace.

1.6 Release History

Table 1-2 lists the major milestones of AMPL Control Configuration commands.

Table 1-2. Release History of the AMPL Control Configuration Commands

Version	Description
APB* 2.4/0	The AMPL Control Configuration Functions is released and delivered together with and as part of Application Builder. This is the first released version of the AMPL Control Configuration functions. It belongs to the AMPL Control Configuration 1.4 product.
APB* 2.5/0	Features like creation of circuit, type circuit, and node control software aspects have been added.
APB* 2.6/0	New command Locate Circuit New view Circuit Usage List
APB* 2.7/0	Export and Import of Advant Objects New command Initialize on Node Control Software aspect New command Update FCB Data on Project OCS Defaults aspect Default logfile - handling Type circuit reference - handling Instance names with absolute reference designation Modify PC page break

Table 1-3 lists the versions of the present user's guide.

Table 1-3. Release History of this User's Guide

Version	Description
3BSE 013 931R0001	First version belonging to the released product.
3BSE 013 931R0101	Describes the control configuration commands for Structure Builder of AMPL Control Configuration 1.5
3BSE 013 931R201	Describes the control configuration commands for Structure Builder of AMPL Control Configuration 1.6
3BSE 013 931R301	Describes the control configuration commands for Structure Builder of AMPL Control Configuration 1.7

1.7 Terminology

The following is a list of terms you should be familiar with.

Term	Description
APB	Abbreviation for Application Builder.
CI	Abbreviation for circuit.
Circuit aspect	Data that describes the control functionality (or a part of the control functionality) of an Advant object to be loaded into the OCS. It is implemented by a circuit object in terms of the AMPL Control Configuration.
DB Element aspect	Data that describes input/output functionality of an Advant object to be loaded into the OCS. The Advant object having a DB Element Aspect can represent either a board or a signal assigned to or still to be assigned to a channel of a board). Within the library it is implemented by a type circuit object in terms of AMPL Control Configuration. Outside the library it is implemented by a specification for an instantiated type circuit.
DB Section aspect	Data that describes input/output functionality of an Advant object to be loaded into the OCS. The Advant object having a DB Section Aspect represents a group of boards with channel - signal assignments done in the OCS structure. The DB Section aspect is implemented by a circuit object in terms of AMPL Control Configuration which holds the instantiated type circuit of the DB Element aspects of the group of boards/signals represented by the Advant object having the DB Section aspect.
Advant object	A data model for an entity treated in design, engineering, operation, and maintenance process.
FCB	Abbreviation for Function Chart Builder.
MDI	Abbreviation for Multi Document Interface.
Node Control Software aspect	Data that represents a node object in terms of the AMPL Control Configuration.
OCS	Abbreviation for Open Control System.
ODB	Internal data repository of Function Chart Builder.
OLE	Abbreviation for Object Linking and Embedding.
ONB	Abbreviation for On-line Builder.

Term	Description
Project OCS Defaults aspect	Data that describes the settings the tools of the AMPL Control Configuration will use by default in the project. Furthermore this unit is implemented by a project object in terms of the AMPL Control Configuration.
STB	Abbreviation for Structure Builder.
TC	Abbreviation for type circuit.
Type Circuit aspect	Data that describes the control functionality (or a part of the control functionality) of an Advant object to be loaded into the OCS as a type. It is implemented by a Type Circuit object in terms of the AMPL Control Configuration.

1.8 Product Overview

AMPL Control Configuration deals with creation, deletion, modification, and documentation of control related aspects of Advant objects and administration of configuration projects.

AMPL Control Configuration comprises the functionality of Application Builder, Function Chart Builder, Bus Configuration Builder, On-line Builder, and additional integration functionality to support interactive working with these tools in the Advant object concept based engineering system.

The functions covered by AMPL Control Configuration are summarized in Table 1-4:

- The available functions depend on the type of the selected aspect in the Advant object.
- In the second column of the Table 1-4 the commands are listed at first that you activate by corresponding buttons in the **Object Window** of Structure Builder. Buttons not invoking any function of AMPL Control Configuration Functions are shown in parentheses.
- Then the commands are listed that are presented in a menu when you activate the **More** button in the **Object Window** window of Structure Builder.
- In case of Circuit aspect type, the **Insert** function is available via drag and drop by mouse.

Table 1-4. Functions Integrated for Different Advant Object Aspects

Aspect Types	Integrated Functions
Project OCS Defaults	New, (Edit,) Delete, Open, (Close,) More: Open, Print, Attach, Set Options, Update Parameter, Generate DB, Update FCB Data
Type Circuit	New, (Edit,) Delete, Open, (Close,) More: Open, Print, Attach, Lock/Unlock
DB Element	New, (Edit,) Delete, Open, (Close,) More: Open, Print, Attach, Update Parameter, Generate DB, Initialize, Lock/Unlock
Circuit	New, (Edit,) Delete, Open, (Close,) More: Open, Print, Attach, Update Parameter, Initialize, Lock/Unlock; Instantiate (by drag and drop), Locate Circuit
DB Section	New, (Edit,) Delete, Open, (Close,) More: Open, Print, Attach, Update Parameter, Generate DB, Initialize, Lock/Unlock
Node Control Software	New, (Edit,) Delete, Open, (Close,) More: Open, Print, Attach, Update Parameter, Convert, Extract, Split, Concatenate, Lock/Unlock, Locate Circuit, Initialize

1.9 User Interface

The functions of AMPL Control Configuration are implemented by Windows NT application programs with the typical graphic user interface and look-and-feel of Windows NT programs. You choose the commands by accelerator keys and shortcuts on the keyboard or by mouse actions.

See detailed information on the graphic user interface and on interacting with Windows NT programs in the user’s guide of Windows NT, for example, *Basics and Installation Microsoft Windows NT Workstation Version 4.0, Microsoft Corporation, Document No. 94945-0996* or select **Help** from the **Start** menu of Windows NT and then look for relevant topics in Windows NT Help.

Chapter 2 Installation

See *Advant Engineering Workplace Release Notes* and *AMPL Control Configuration Release Notes*.

Chapter 3 Configuration/Application Building

3.1 Considerations

3.1.1 About the Advant Object Technology

AdvaBuild Engineering Software is based on the **Advant object** technology:

- Real world objects – physical objects or virtual ones such as functions – are abstracted to **Advant objects**. An Advant object is a data model carrying all the information adherent to an entity treated in the design, engineering, operation, and maintenance process.
- The data of an Advant object are grouped into **aspects**. An aspect is an assembly of data describing certain properties of the Advant object.
- The data of an Advant object can be presented in several specific ways: aspects can have several **views**.
- Aspect data are handled and views are presented by the different **builders** or the Control Configuration Functions described in this manual.
- The Advant Engineering Workplace and the AMPL Control Configuration products comprise all of these cooperating tools supporting engineering tasks of projects to be implemented with Advant OCS.

For a general description of the Advant object technology, see the *Structure Builder User's Guide*.

In this Section 3.1, Considerations we give an overview of aspects relevant for OCS control configuration. These aspects are available via the functions of the product AMPL Control Configuration. We show access to these functions and the general ways of activating them. Finally, you can read about considerations on parameter references, DB element handling, and circuit handling.

3.1.2 Aspects for OCS Control Configuration

Among the numerous aspects of Advant objects we concentrate on those relevant for OCS control configuration.

The following table lists these aspects together with a short description of aspect relevant data and Advant object types, for which the aspects are applicable:

As usual, each aspect can have different views.

Table 3-1. Aspects for OCS Control Configuration

Aspect	Aspect Data Contents	For Standard Object Types
Project OCS Defaults	Project defaults and other data describing an OCS programming project	Project
Type Circuit	Type Circuit data	Object type used to represent parameterized type; typically for building Process Objects
DB Element	Type Circuit data (The DB part of TC creates a board DB element and all signal DB elements of fitting type.)	Board
	Type Circuit data (The DB part of TC modifies a signal DB element of fitting type.)	Signal
	Type Circuit data (The DB part of TC creates or modifies general purpose DB elements.)	Object type used to represent general purpose DB elements
Circuit	Circuit data	Process Object
DB Section	Collection of DB Element aspects	Node (or its part) in OCS structure
Node Control Software	Node data	Node
Parameter	Parameter data for Process Objects or objects representing DB elements	Various

3.1.3 Parameter References

A parameter reference specified in a certain aspect describes from which other aspect and attribute to get the actual value for the parameter.

You can define parameter references for symbols of type circuits

- To be instantiated into the circuit of a Circuit aspect
- Or to be used or referenced in a DB Element aspect.

You can configure parameter references with the symbol editor in type circuit mode of Function Chart Builder.

Parameter references are executed with the following commands:

- **Update Parameter**
- **Generate DB**
- **Locate Circuit**
- **Drag and drop** (object containing Circuit aspect).

Parameter references can be of one of the following types:

- Relative reference to the same Advant object
- Relative reference to another Advant object
- Absolute reference to an Advant object.

3.1.4 DB Element Handling

You can represent single board and signal DB elements or other kinds of DB elements as aspects of type DB Element of Advant objects representing a board or a signal or another entity. Data behind the DB Element aspect is a type circuit specifying a board DB element, a signal DB element, or another kind of DB element. These type circuits can include parameter references, typically relative parameter references to the same Advant object.

A board type circuit:

- References to the absolute reference designation (used to define the target system item designation for the DB element) within the OCS structure
- References to the basic property name of category Customer ID if the corresponding board DB element has a name terminal
- References to parameter aspects (if available for the board).

A signal type circuit:

- References to the absolute reference designation (used to define the target system item designation for the DB element) within the OCS structure
- References to the basic property name of category Customer ID
- References to the basic property description of category Customer ID
- References to parameter aspects.

If for each type of board, signal, or other entity you introduce an Advant object with a structure as described above in a library (project). You can then instantiate these Advant objects by copying them from the library to your project. The DB element aspects of the instance Advant objects only reference the type circuits of the DB element aspects of the library Advant objects.

All or a certain amount of DB elements of a node in a project can be represented by an aspect of type DB Section of an Advant object.

The data of a DB Section aspect is accumulated from the data of the aspects of type DB Element in the control system oriented OCS structure below the Advant object carrying the DB Section aspect. Accumulation is done using the **Generate DB** command. Destination of the **Generate DB** command is either the circuit of the DB Section aspect or the node data of the Node Control Software aspect of a node object the DB Section belongs to.

The **Generate DB** command instantiates all DB element type circuits contained in the substructure below the Advant object carrying the DB Section aspect into the destination.

Prerequisite for using this command is that I/O-planning (building up the control system oriented OCS structure) has been done.

The command takes care of all data values accessed via parameter references:

- Item designation and name (determined by I/O-planning)
- Parameter values.

After changes in parameter aspects you can update the parameters in the instantiated DB element type circuits the node by using the **Update Parameter** command on the corresponding DB Section aspect.

3.1.5 Circuit Handling

You can represent the control and/or the I/O functionality of physical or virtual objects by aspects of type Circuit of Advant objects (e.g. of standard object type Process Object) representing these physical or virtual objects.

Data behind the Circuit aspect is a circuit with at least one instantiated type circuit specifying control functionality. The type circuit can include parameter references typically relative parameter references to the same Advant Object:

- Reference to the basic property name of category Customer ID, for example, for naming purposes
- References to parameter aspects.

If for each type of Process Object you introduce an Advant object with a structure as described above in a library (project), you can then instantiate these objects by copying them from the library to your project. The Circuit aspects of the instance objects only reference the type circuits of the Circuit aspects of the library objects.

You can insert the instance Process objects in your project into the node data behind the Node Control Software aspect of a node by drag and drop from Structure Builder to the Function Chart or Tree representation of a PC program of the node. During drop the circuit is expanded, the parameters are updated and the circuit contents is shown in Function Chart and Tree representation.

After changes in parameter aspects you can update the parameters in the inserted objects in the node by using the **Update Parameter** command on the corresponding Circuit aspect.

3.2 Capacity and Performance

Restrictions on Parallel Execution

The following AMPL Control Configuration components may be started only once on one client PC:

- Application Builder
- On-line Builder
- Bus Configuration Builder.

Function Chart Builder may be started more than one time on different objects on the same client PC.

3.3 Application Start-up

For start-up of Structure Builder, see *Structure Builder User's Guide*.

You can start-up the functions of the AMPL Control Configuration in any of the following ways:

- By activating Structure Builder commands on Advant objects containing OCS Control Configuration aspects (for example, **Edit | Copy and Edit | Paste Special**)
- By activating commands of the **Object Window** on OCS Control Configuration aspects of an Advant object
- By activating commands of a selected control configuration aspect in the Aspect window part of a structure or pool window.

Chapter 4 Runtime Operation

4.1 Operating Overview

You can access AMPL Control Configuration functions by Structure Builder object and aspect commands. Where necessary, the AMPL Control Configuration functions provide different views on the data of an aspect of a certain type. In this case you can use the aspect commands on a specified view. Here, we give you an overview of the command usage.

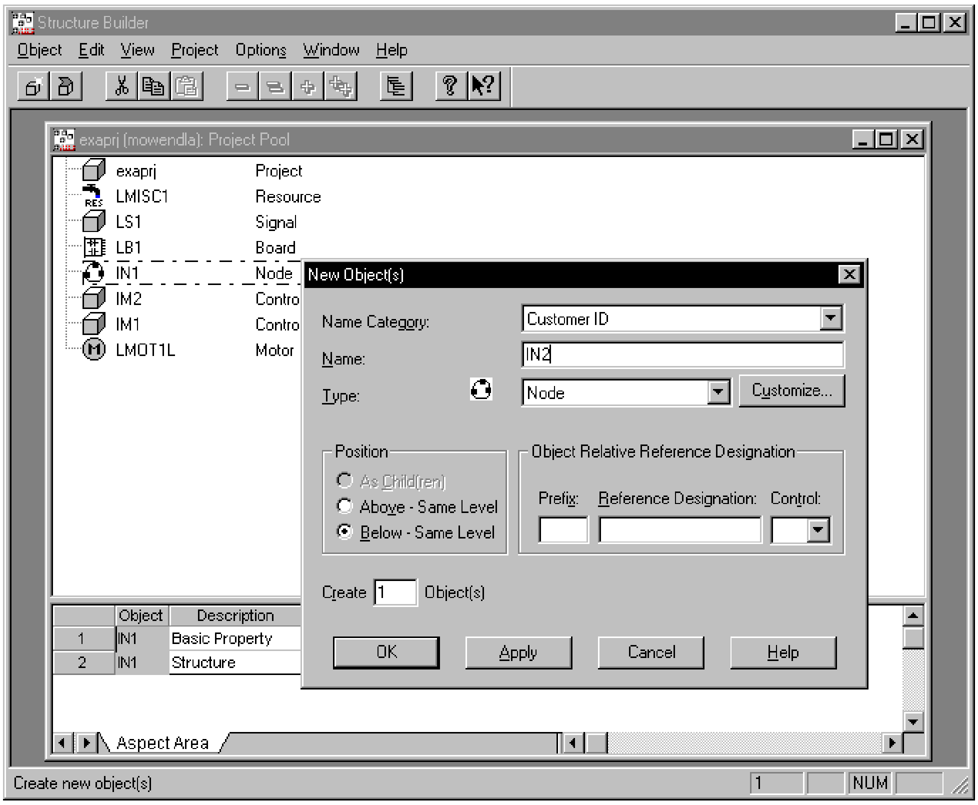
4.1.1 Create (New) an Advant Object

You can introduce a new Advant object representing a project with the **Project | Administration** command.

You can create a new Advant object representing a physical object in the project's Advant object pool with the **Object | New** command.

The Advant object pool is represented in the **Project Pool** window.

With these commands you define the basic properties aspect of the Advant object.


Structure Builder
Object Edit View Project Options Window Help
exaprj (mowendla): Project Pool
New Object(s)
Customize...
OK
Apply
Cancel
Help
Aspect Area
NUM

exaprj	Project
LMISC1	Resource
LS1	Signal
LB1	Board
IN1	Node
IM2	Contro
IM1	Contro
LMOT1L	Motor

Name Category: Customer ID

Name: IN2

Type: Node

Position
○ As Child(ren)
○ Above - Same Level
● Below - Same Level

Object Relative Reference Designation
Prefix: Reference Designation: Control:

Create 1 Object(s)

	Object	Description
1	IN1	Basic Property
2	IN1	Structure

Create new object(s)

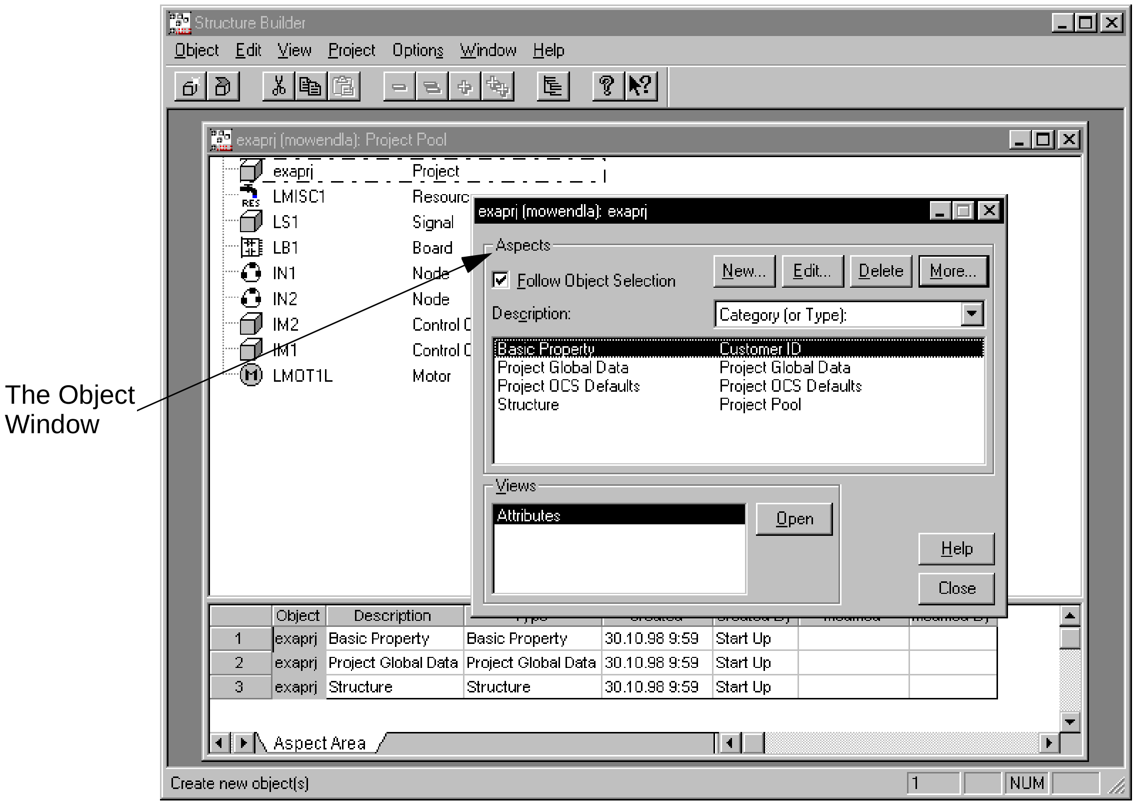
Figure 4-1. Project Pool Window

4.1.2 Open an Advant Object

You can open an existing Advant object with the **Object | Open** command.

If you choose this command, Structure Builder shows the **Object Window**.

The **Object Window** lists all existing aspects of an Advant object and allows to use commands on the aspects.


The Object Window
Structure Builder
Object Edit View Project Options Window Help
exaprj (mowendla): Project Pool
exaprj (mowendla): exaprj
Aspects
New... Edit... Delete More...
Views
Open
Help
Close
Aspect Area

exaprj	Project
LMISC1	Resourc
LS1	Signal
LB1	Board
IN1	Node
IN2	Node
IM2	Control C
IM1	Control C
LMOT1L	Motor

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
Project Global Data	Project Global Data
Project OCS Defaults	Project OCS Defaults
Structure	Project Pool

Attributes

	Object	Description					
1	exaprj	Basic Property	Basic Property	30.10.98 9:59	Start Up		
2	exaprj	Project Global Data	Project Global Data	30.10.98 9:59	Start Up		
3	exaprj	Structure	Structure	30.10.98 9:59	Start Up		

Create new object(s) 1 NUM

Figure 4-2. Object Window

Additionally, the Aspect window is included as split window in each structure window such as the **Project Pool** window. It shows all aspects of a selected amount of Advant objects, and is therefore called the **Aspect Area**.

4.1.3 Delete an Advant Object

You can delete an existing Advant object with the **Object | Delete** command. This command leads to deleting all aspects of the Advant object.

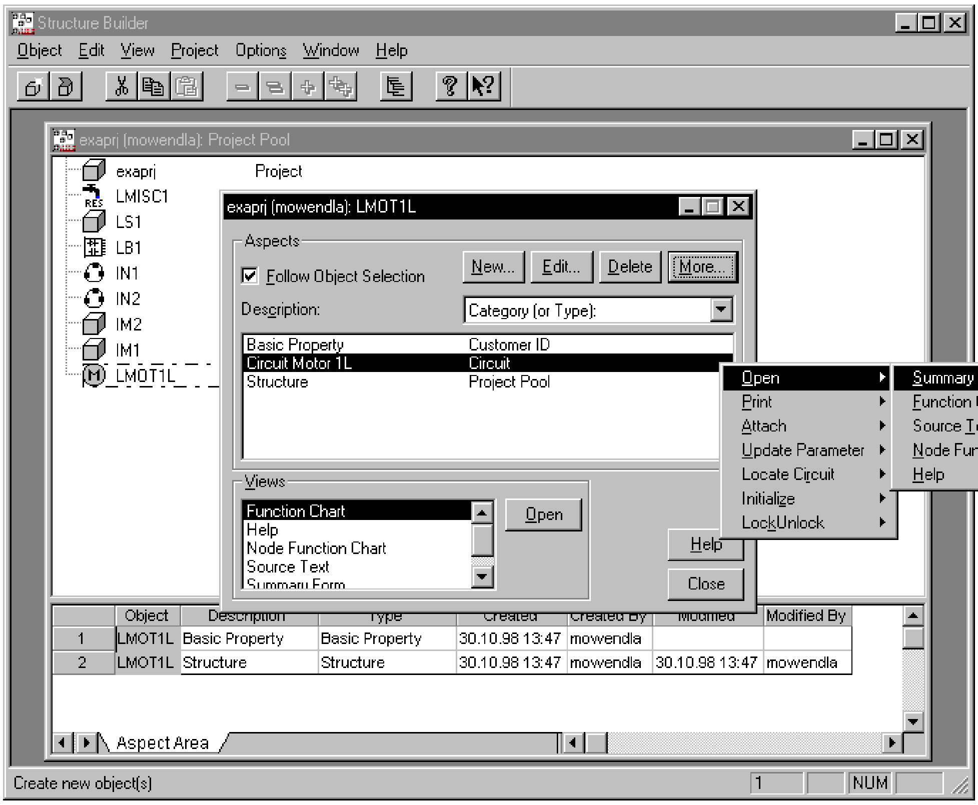
4.1.4 Copy an Advant Object

You can copy an existing Advant object available in the same or another project by the **Edit | Copy** and **Edit | Paste (Special)** command sequence or by drag and drop to the same or another project.

Performing such a command leads to copying all the aspects of the source Advant object you have chosen to be copied.

4.1.5 Buttons and Popup Menu in Object Window

The commands for a selected aspect are either provided by buttons in the **Object Window** or by menu items of the command popup menu which you can display by activating the **More** button or by clicking on the right mouse button.



Description:	Category (or Type):
Basic Property	Customer ID
Circuit Motor 1L	Circuit
Structure	Project Pool

	Object	Description	Type	Created	Created By	Modified	Modified By
1	LMOT1L	Basic Property	Basic Property	30.10.98 13:47	mowendla		
2	LMOT1L	Structure	Structure	30.10.98 13:47	mowendla	30.10.98 13:47	mowendla

Figure 4-3. Object Window with Command Popup Menu

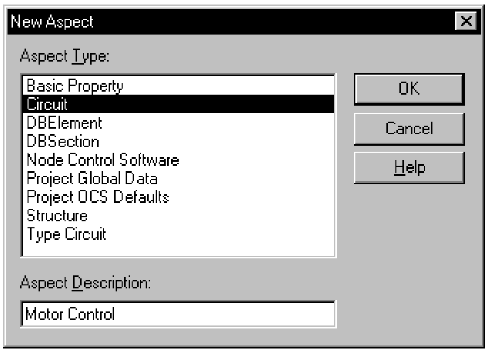
1. Select at first an aspect in the **Object Window** and activate the **More** button.
2. Then choose a command in the popup menu.
3. Finally, if applicable, on the last level you may open a view.

The scope of the commands depends on their context.

4.1.6 New Aspect

In the **Object Window**, you can create a new aspect by activating the **New** button.

Structure Builder opens the **New Aspect** dialog where you can define the type of the new aspect.



Aspect Type: Circuit

Aspect Description: Motor Control

Figure 4-4. New Aspect Dialog

4.1.7 Open Aspect

From within the **Object Window** you can open and edit an aspect with the **Open** button or with the **Open** command from the command popup menu.

4.1.8 Print Aspect

From within the **Object Window** you can print an aspect with the **Print** command from the command popup menu.

4.1.9 Delete Aspect

From within the **Object Window** you can delete an aspect with the **Delete** button or the **Delete** command from the command popup menu.

4.1.10 Additional Commands

AMPL Control Configuration registers additional commands and command views for certain aspect types. These commands are offered in the command popup menu.

Examples

For aspect type Node Control Software the **Convert**, **Split**, **Extract** commands are additionally offered.

For aspect type Project OCS Defaults the **Set Options** command is additionally offered.

4.2 Runtime Tutorial

To become familiar with the way OCS Control Configuration functions are integrated into Structure Builder you are encouraged to work through the following sequence of exercises.

1. Create a new project (a new project Advant object) with Structure Builder.
   (Login as user `eostb`, and so on.
   See *Structure Builder User's Guide* and *Engineering Methods Reference Manual*.)
2. Login as normal user (member of group `eoeng`) again and do the following steps described in Section 4.3, Operating Instructions.
   Project OCS Defaults aspect will be automatically created for the project Advant object. (This aspect represents the project parts of the AMPL Control Configuration and is necessary to be available before working on other OCS Control Configuration aspects.)
3. Create Advant objects representing physical objects (for example, motors) as instances in this project or copy them from another (library) project.
4. Create new or copy Circuit aspects to the Advant objects.
5. If not copied: Edit (including use of TCs) or attach the contents (the CIs) of each single Circuit aspect.
6. Create an Advant object representing a node/controller
7. Add a Node Control Software aspect to the node Advant object.
8. Edit the node including use of CIs and TCs. Or attach the node.
9. Try further functions of the components of the AMPL Control Configuration product.

4.3 Operating Instructions

4.3.1 Set Options, Project OCS Defaults Aspect

Configure the options for working with AMPL Control Configuration functions using the **Set Options** command and dialog on the Project OCS Defaults aspect.

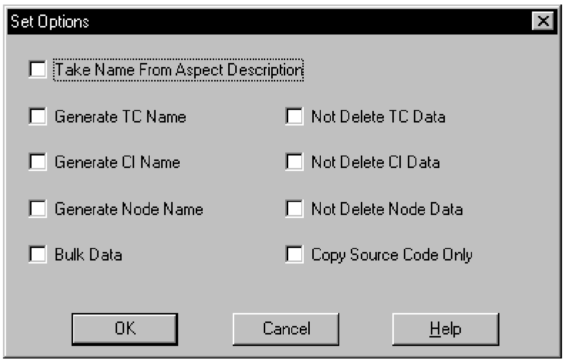


Figure 4-5. Set Options Dialog

Set the options by checking the check boxes of the **Set Options** dialog.

Take Name From Aspect Description

If you select this option, the name proposal displayed in the **New Type Circuit** or **New Circuit** or **New Node** dialog is taken from the first eight characters of the aspect description.

By default (not checked) name proposal displayed is taken from the first eight characters of basic property name of category Customer ID.

Generate TC Name

If you select this option, the name proposal displayed in the **New Type Circuit** and **Copy** dialogs is composed from the normal (max. eight characters) name proposal and a number, increased by one for each instance, added to or overwriting the last characters of this initial name proposal.

Table 4-1. Generated TC Names (Example)

Basic Property Name of Category Customer ID	Generated Names
TC_MOT	TC_MOT1
	TC_MOT2
	TC_MOT3
	…
	TC_MO143

Select this option when you want to do a bulk copy of objects with generated name for the new type circuits.

By default (not checked) no number is included.

Generate CI Name

If you select this option, the name proposal displayed in the **New Circuit** and **Copy** dialogs is composed from the normal (max. eight characters) name proposal and a number, increased by one for each instance, added to or overwriting the last characters of this initial name proposal.

Table 4-2. Generated CI Names (Example)

Basic Property Name of Category Customer ID	Generated Names
CI_MOT	CI_MOT1
	CI_MOT2
	CI_MOT3
	…
	CI_MO143

Select this option when you want to do a bulk copy of objects, for example, import of a consumer list with generated name for the new circuits.
By default (not checked) no number is included.

Generate Node Name

If you select this option, the name proposal displayed in the **New Node** and **Copy** dialogs is composed from the normal (max. eight characters) name proposal and a number, increased by one for each instance, added to or overwriting the last characters of this initial name proposal.

Table 4-3. Generated Node Names (Example)

Basic Property Name of Category Customer ID	Generated Names
NODE_A	NODE_A1
	NODE_A2
	NODE_A3
	…
	NODE_143

Select this option when you want to do a bulk copy of objects with generated name for the new nodes. By default (not checked) no number is included.

Bulk Data

If you select this option, the **Copy** dialog of the paste action or **Attach** command is not shown but the operation is performed with the name proposal immediately without the need for dialog confirmation.
Select this option before you want to perform a bulk copy of objects, for example, the import of a Consumer list or Signal list.
By default (not checked) the **Copy** dialog is shown and activation of the **OK** button is necessary.

Not Delete TC Data

If you select this option, the **Delete** command does not delete the type circuit data represented by the Type Circuit or DB Element aspect. It only removes the assignment of the data to the corresponding aspect of the Advant object. By default (not checked) type circuit data are deleted.

Not Delete CI Data

If you select this option, the **Delete** command does not delete the circuit data represented by the Circuit or DB Section aspect. It only removes the assignment of the data to the corresponding aspect of the Advant object. By default (not checked) circuit data are deleted.

Not Delete Node Data

If you select this option, the **Delete** command does not delete the node data represented by the Node Control Software aspect. It only removes the assignment of the data to the corresponding aspect of the Advant object. By default (not checked) node data are deleted.

Copy Source Code Only

If you select this option, the copy/paste action or **Attach** command does only copy the source code of the node, type circuit, or circuit data. By default (not checked) all data are copied.

Remark

We recommend you to set the following options for the target project by using the **Set Options** command on the target project's Project OCS Defaults aspect before doing a bulk copy of objects from library project to another target project (for example, before import or before performing manually a copy with multiple selection):

Bulk Data, **Generate TC Name**, **Generate CI Name**, **Generate Node Name**.

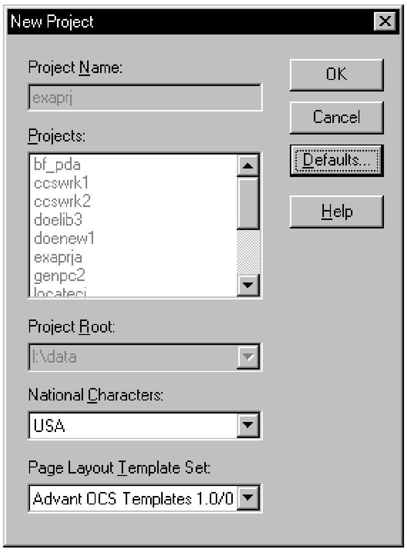
This switches off the **Copy** dialog for the aspects owned by the Control Configuration Functions during the bulk copy process and generates file-object names automatically. Switching on the **Generate Name** options is only needed if the destination project already can contain instances with the same names as in the source project.

4.3.2 Create Project OCS Defaults Aspect

This aspect represents a subproject of the AMPL Control Configuration functions within a project administered with Structure Builder.

The Project OCS Default aspect of a project object is created automatically during creation of the project. It cannot be created manually.

During project creation the **New Project** dialog of Application Builder is displayed.



Project Name: exaprj

Projects:
- bf_pda
- ccswrk1
- ccswrk2
- doelib3
- doenew1
- exaprja
- genpc2

Project Root: I:\data

National Characters: USA

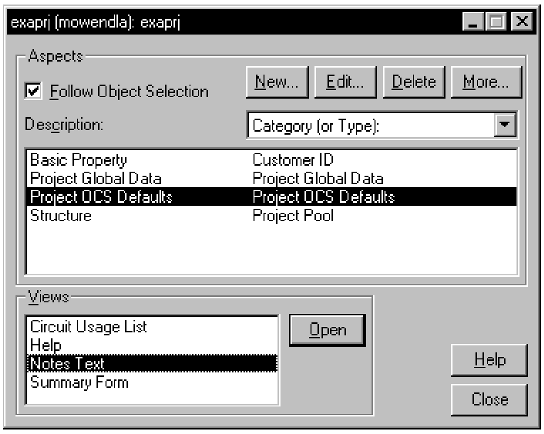
Page Layout Template Set: Advant OCS Templates 1.0/0

Figure 4-6. New Project Dialog

The project name and the project root directory are fixed. The project name equals the project name defined with the **Project | Administration** command of Structure Builder.

You can select the national character set and template set and you can use the subdialogs for setting the project defaults.

After activation of the **OK** button in the **New Project** dialog the **Object Window** shows the added Project OCS Defaults aspect.



☑ Follow Object Selection

Description:	Category (or Type):
Basic Property	Customer ID
Project Global Data	Project Global Data
Project OCS Defaults	Project OCS Defaults
Structure	Project Pool

Views:
- Circuit Usage List
- Help
- Notes Text
- Summary Form

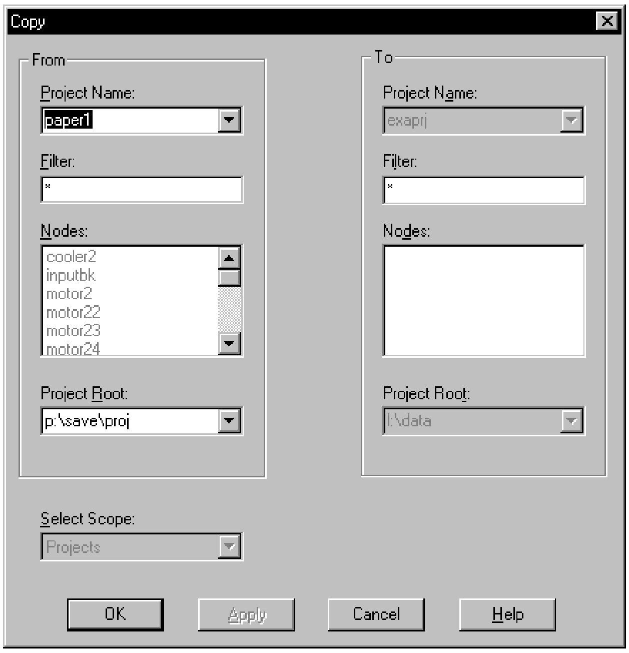
Figure 4-7. Object Window Showing Project OCS Defaults Aspect

4.3.3 Attach Project OCS Defaults Aspect

You can attach the project defaults data of an existing subproject to a selected Project OCS Defaults aspect by using the **Attach** command from the command popup menu of the aspect within the **Object Window**.

Existing project defaults data (project defaults and project notes) on ABB Master standard directories (either represented by a Project OCS Defaults aspect in an project Advant object or not) can be used.

The **Attach** command shows the **Copy** dialog of Application Builder.



From

Project Name: paper1

Filter: *

Nodes:
cooler2
inputbk
motor2
motor22
motor23
motor24

Project Root: p:\save\proj

To

Project Name: exaprj

Filter: *

Nodes:

Project Root: l:\data

Select Scope: Projects

Figure 4-8. Copy Dialog for Attach Project OCS Default Aspect

The **Copy** dialog is shown with scope fixed to Projects, **To** section fixed to the current project and project root, **To** (project-) name in the name entry subdialog fixed to the current (project-) name, multiple selection in the **From** section disabled.

Before activating the **OK** button you select the source project (which results in the source project name in the name entry subdialog) and project root in the **From** section of the dialog.

After performing the **Copy** operation project defaults data (project defaults and project notes) for the subproject are available.

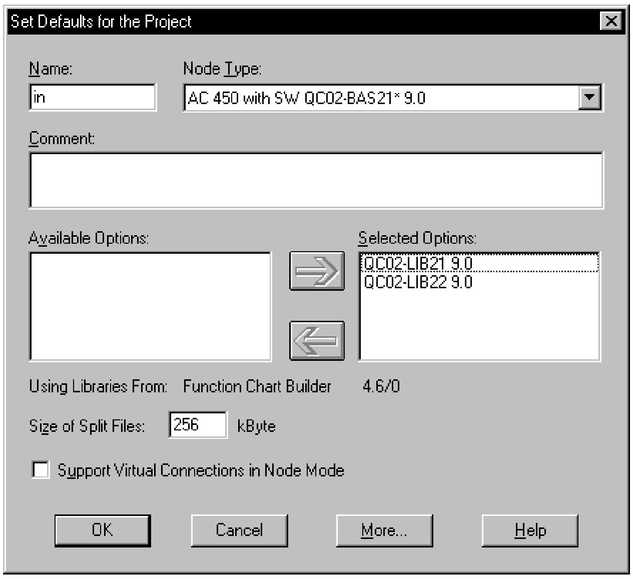
4.3.4 Open Project OCS Defaults Aspect

After you have selected a Project OCS Defaults aspect of an Advant object representing a project and a certain view within the **Object Window** you can open it using the **Open** button.

Another possibility for you to open the Project OCS Defaults aspect is to display the aspect’s command popup menu with the **More** button, select the **Open** command and activate the appropriate view from the cascaded view menu.

View Summary Form

Open shows the **Set Project Defaults** dialog. This dialog allows you to edit the project defaults.



Name: in

Node Type: AC 450 with SW QC02-BAS21* 9.0

Comment:

Available Options:

Selected Options:
QC02-LIB21 9.0
QC02-LIB22 9.0

Using Libraries From: Function Chart Builder 4.6/0

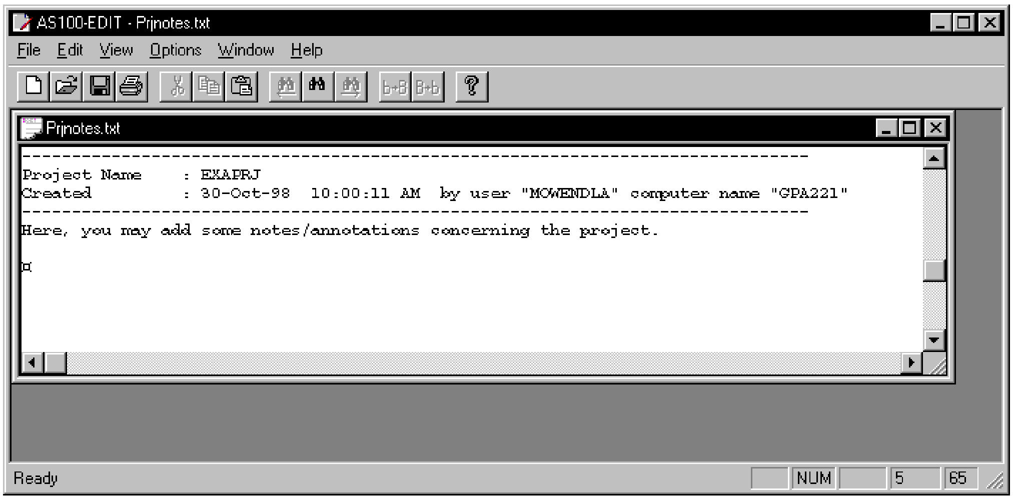
Size of Split Files: 256 kByte

☐ Support Virtual Connections in Node Mode

Figure 4-9. Open View Summary Form on Project OCS Defaults Aspect

View Notes Text

Open starts up the **AS100-Edit** function with the PRJNOTES.TXT file.



```
------------------------------------------------------------------------------
Project Name     : EXAPRJ
Created          : 30-Oct-98  10:00:11 AM  by user "MOWENDLA" computer name "GPA221"
------------------------------------------------------------------------------
Here, you may add some notes/annotations concerning the project.

¤
```

Figure 4-10. Open View Notes Text on Project OCS Defaults Aspect

4.3.5 Delete Project OCS Defaults Aspect

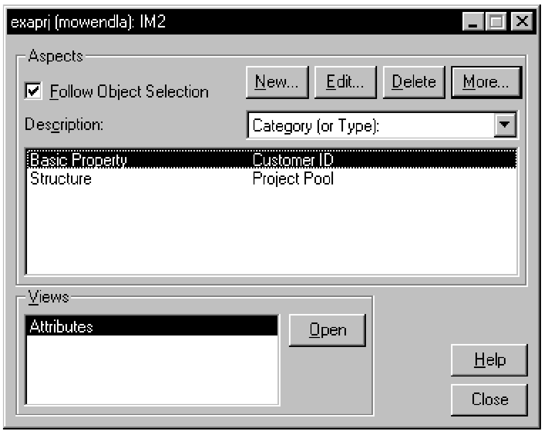
You cannot delete the Project OCS Defaults aspect. It is deleted when the project is deleted with Structure Builder.

4.3.6 Create Aspect

You can create the following new aspects with the **New** button and **New Aspect** dialog of the **Object Window**:

- Type Circuit / Circuit aspect for an Advant object representing a physical object
- DB Element aspect for an Advant object representing a board or signal or other kind of DB element
- DB Section aspect for an Advant object representing the DB elements of the node or part of a node
- Node Control Software aspect for an Advant object representing an OCS node.

DB Element aspect is mapped to a type circuit and DB Section aspect is mapped to a circuit in terms of the Application Builder and Function Chart Builder contained in the Control Configuration Functions.



exapri (mowendla): IM2

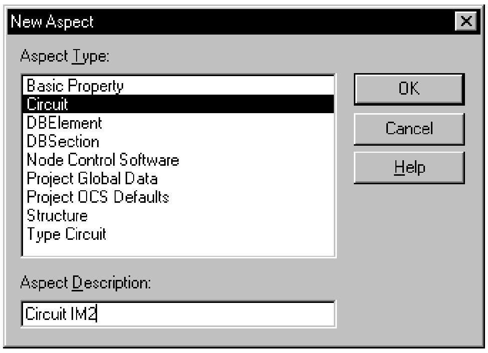
☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
Structure	Project Pool

Figure 4-11. Initial Object Window

After you pressed the **New** button the **New Aspect** dialog is displayed.



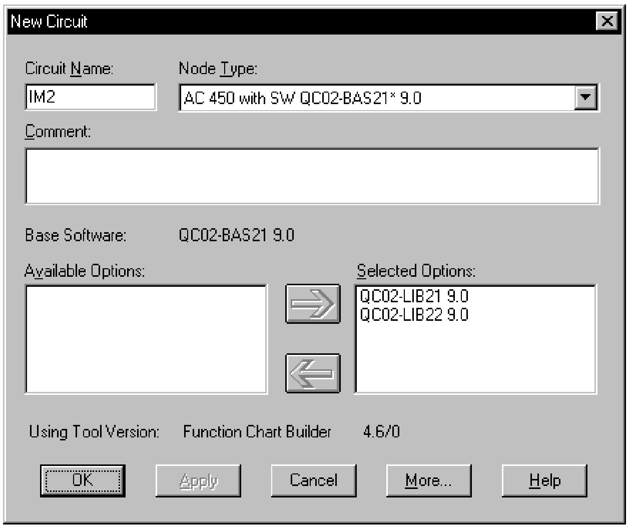
Aspect Type:

- Basic Property
- Circuit
- DBElement
- DBSection
- Node Control Software
- Project Global Data
- Project OCS Defaults
- Structure
- Type Circuit

Aspect Description: Circuit IM2

Figure 4-12. New Circuit Aspect

After you pressed the **OK** button of the **New Aspect** dialog, you may define the type circuit or circuit in terms of AMPL Control Configuration using the **New Type Circuit** or **New Circuit** dialog.



Circuit Name: IM2

Node Type: AC 450 with SW QC02-BAS21* 9.0

Comment:

Base Software: QC02-BAS21 9.0

Selected Options:
QC02-LIB21 9.0
QC02-LIB22 9.0

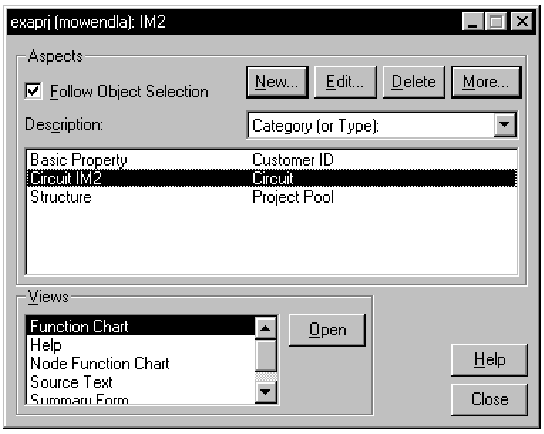
Using Tool Version: Function Chart Builder 4.6/0

Figure 4-13. New Circuit Dialog

By default the first eight characters of the Advant object default basic property name (of category Customer ID) are offered as type circuit / circuit name.

You can choose to overwrite the proposed name of the new type circuit or circuit or to keep the proposed name and you can enter additional configuration data

After you pressed the **OK** button of the **New Type Circuit / New Circuit** dialog, the **Object Window** shows the created aspect.



Description	Category (or Type):
Basic Property	Customer ID
Circuit IM2	Circuit
Structure	Project Pool

Function Chart
Help
Node Function Chart
Source Text
Summary Form

Figure 4-14. Object Window with Added Circuit Aspect

4.3.6.1 Type Circuit Reference-Handling

A Type circuit reference can be created by the **Aspect | New** command by entering an already existing type circuit (file) name in the **New Type Circuit** dialog for a DB Element or Type Circuit aspect and acknowledging the following two query messages positively with pressing the Yes button.

1. This aspect already exists. Do you want to register this aspect in Structure Builder?
2. Do you want to create a reference to type circuit <name>?

The reference is taken care of during copying, deleting and exporting and importing the aspects.

NOTE

Up to including ACC 1.6 type circuit references have only been supported for DB Element aspects copied from a library project.

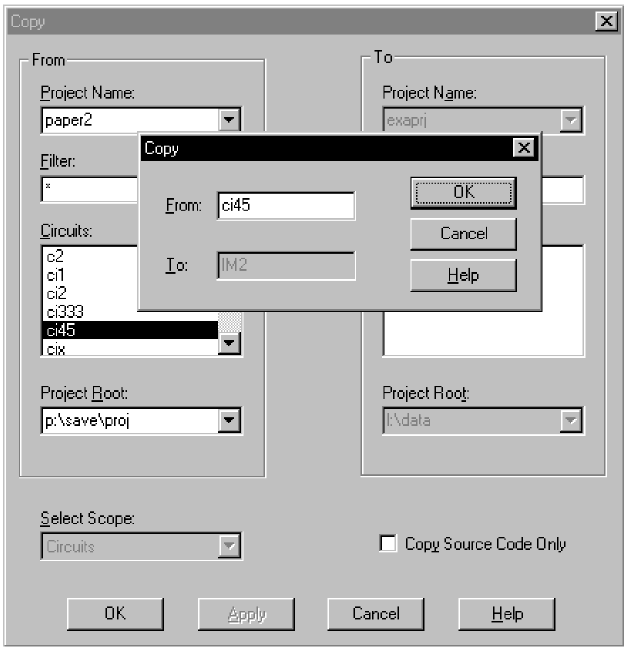
4.3.7 Attach Aspect

You can attach the data of an existing type circuit or circuit to the following selected aspects by using the **Attach** command from the command popup menu on the aspect within the **Object Window**:

- Type Circuit / Circuit aspect for an Advant object representing a physical object
- DB Element aspect for an Advant object representing a board or signal or another kind of DB element
- DB Section aspect for an Advant object representing the boards and assigned signals or other kind of DB element of the node or part of a node.

You can attach a type circuit or circuit existing on known ABB Master standard directories (either represented by a Type Circuit aspect or Circuit aspect in an Advant object or not.)

The **Attach** command shows the **Copy** dialog of Application Builder.



From

Project Name: paper2

Filter: *

Circuits: c2, ci1, ci2, ci333, ci45, cix

Project Root: p:\save\proj

To

Project Name: exaprj

Project Root: I:\data

Select Scope: Circuits

☐ Copy Source Code Only

Copy

From: ci45

To: IM2

Figure 4-15. Copy Dialog for Attach Circuit Aspect

It is shown with scope fixed to Type Circuits or Circuits, **To** section fixed to the current project and project root, **To** name in the name entry subdialog fixed to the current name.

Before activating the **OK** button you can select in the **From** section the source project and project root and you can select the source type circuit or circuit. The **Copy Source Code Only** option can be selected.

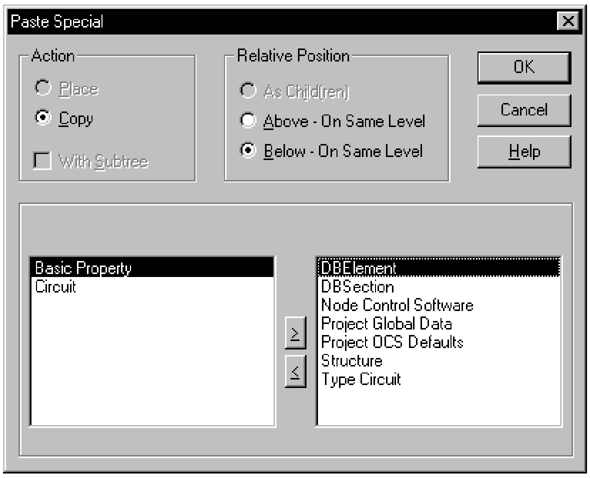
After you pressed the **OK** button of the basic **Copy** dialog the name entry subdialog is displayed. You still can change the **From** name here, but normally you have already selected the right type circuit / circuit in the **From** selection. Press the **OK** button to finalize the **Attach** command’s **Copy** function.

After performing the **Copy** operation all data for the aspect are available as given by the source.

4.3.8 Copy Aspect

You can copy the following new aspects with the **Edit | Copy** and **Edit | Paste (Special)** sequence of object commands when duplicating an object:

- Type Circuit / Circuit aspect for an Advant object representing a physical object
- DB Element aspect for an Advant object representing a board or signal or another kind of DB element
- DB Section aspect for an Advant object representing the boards and assigned signals or other kind of DB element of the node or part of a node.



Paste Special

Action
○ Place
● Copy
☐ With Subtree

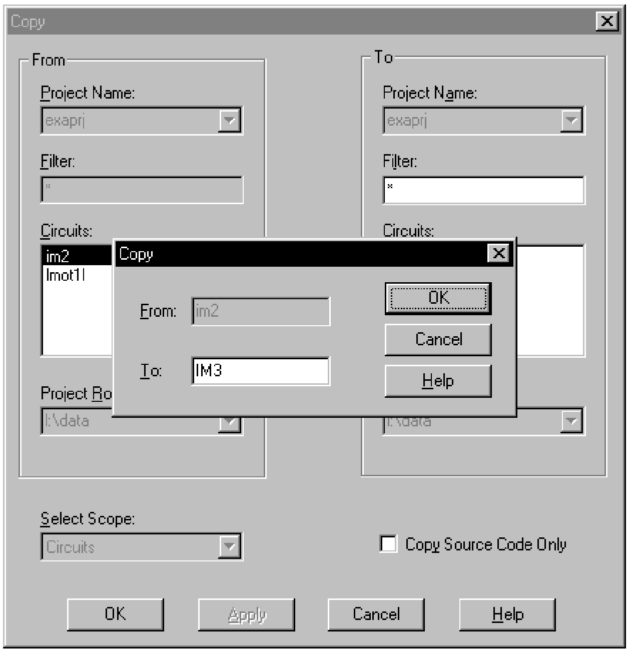
Relative Position
○ As Child(ren)
○ Above - On Same Level
● Below - On Same Level

Basic Property
Circuit

DBElement
DBSection
Node Control Software
Project Global Data
Project OCS Defaults
Structure
Type Circuit

Figure 4-16. Paste Special Dialog for Circuit Aspect

The paste function shows the **Copy** (Type Circuit or Circuit) dialog of Application Builder.



Copy

From
Project Name: exaprj
Filter: *
Circuits:
im2
lmot1l
Project Ro
I:\data

To
Project Name: exaprj
Filter: *
Circuits:
I:\data

Copy
From: im2
To: IM3

Select Scope: Circuits
☐ Copy Source Code Only

Figure 4-17. Copy Dialog for Circuit Aspect

It is shown with scope fixed to Type circuits or Circuits, **From** section fixed to current (source) project, type circuit / circuit name and (source) project root, **To** section fixed to destination project and destination project root. Before activating the **OK** button you can select the **Copy Source Code Only** option, after activating the **OK** button you have to enter the name of the new type circuit or circuit for the new Type Circuit aspect or Circuit aspect in the name entry **Copy** subdialog.

A copy of the source aspect is added to the destination object.

NOTE

In case you copy between different projects, the copy/paste function on a DB Element aspect does not copy the type circuit data but references the source type circuit from the target.

4.3.9 Open Aspect

You can open and edit the following existing aspects using the **Open** button or double clicking on the selected view:

- Type Circuit / Circuit aspect for an Advant object representing a physical object
- DB Element aspect for an Advant object representing a board or signal or another kind of DB element
- DB Section aspect for an Advant object representing the boards and assigned signals or other kind of DB element of the node or part of a node.

Alternatively, the **Open** command of the aspect’s popup menu can be used.

View Function Chart

Open starts up the Function Chart Builder on the type circuit or circuit.

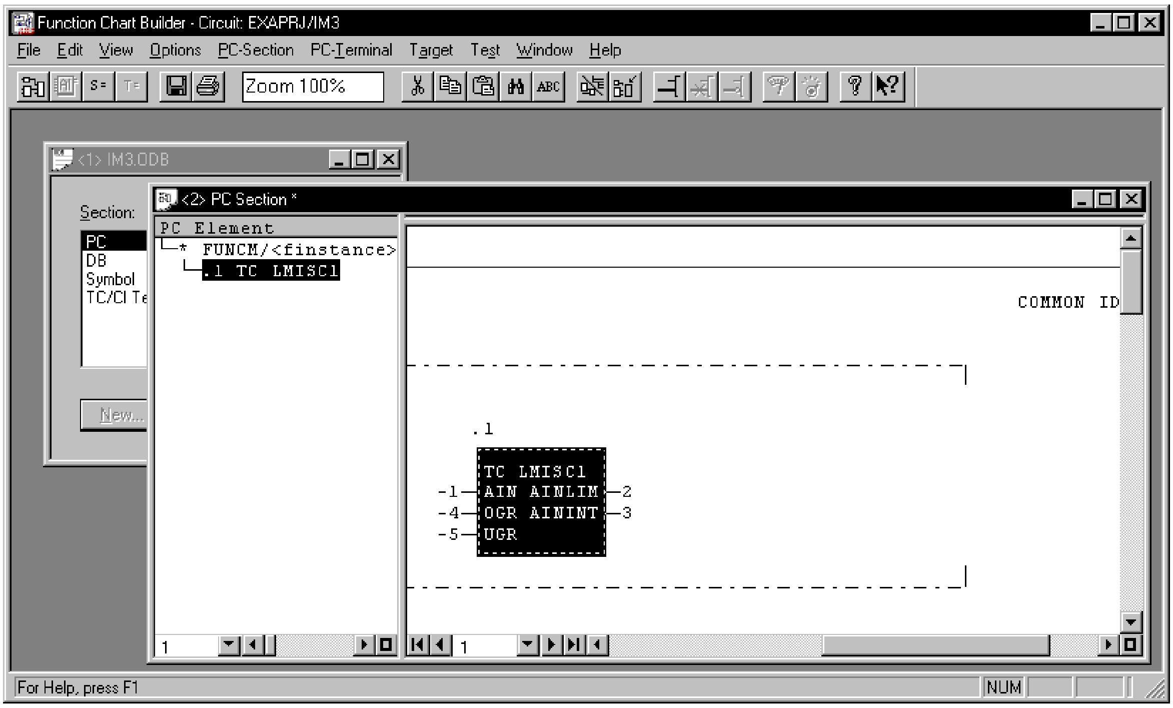


Figure 4-18. Open View Function Chart on Circuit Aspect

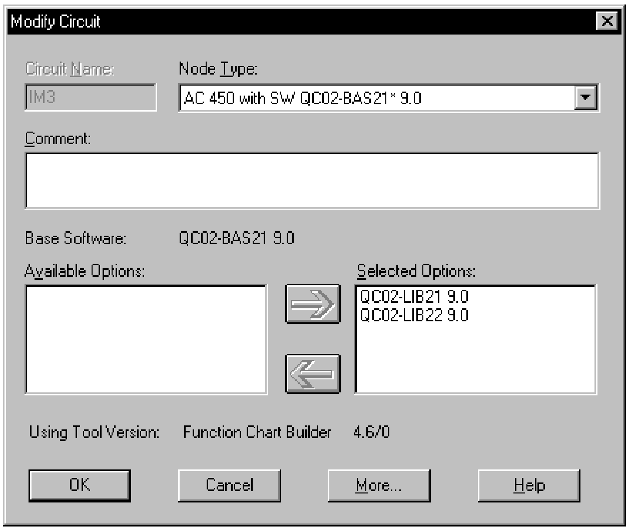
NOTE

> You might have to backtranslate an attached type circuit or circuit (if attached with **Copy Source Code Only** option or just source code available) before you can start to edit it. After editing you should generate the source code using Function Chart Builder's **Generate Source** command.

Editing (and using other Function Chart Builder functions within the application window) on the type circuit or circuit can be done without restrictions compared to today.

View Summary Form

Open starts up dialog **Modify Type Circuit** or **Modify Circuit** where you can change default settings for the type circuit or circuit.



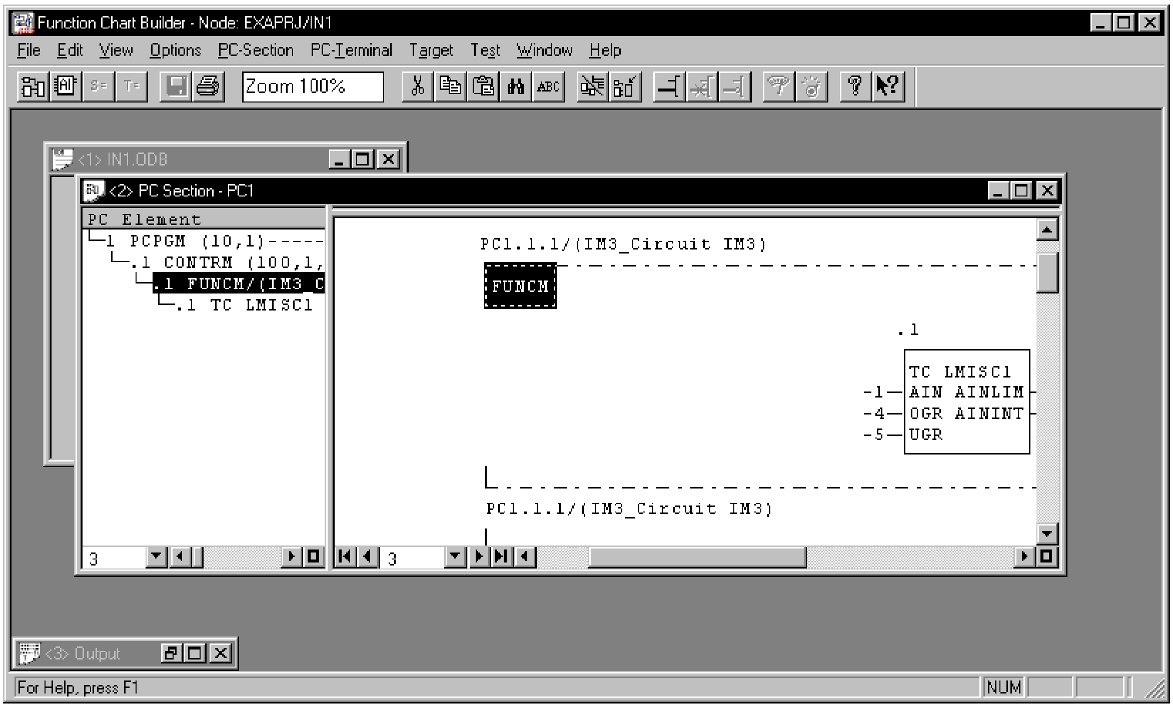
Circuit Name: IM3
Node Type: AC 450 with SW QC02-BAS21* 9.0
Comment:
Base Software: QC02-BAS21 9.0
Selected Options:
QC02-LIB21 9.0
QC02-LIB22 9.0
Using Tool Version: Function Chart Builder 4.6/0

Figure 4-19. Open View Summary Form for Circuit Aspect

View Node Function Chart

Applicable for Circuit aspect, it starts up the Function Chart Builder application window on the node, into which the object has been dropped.

The starting PC element (for example, FUNCM) of the circuit in the PC program of the node is selected and shown in tree or function chart or both.



```
PC Element
└1 PCPGM (10,1)-----
  └.1 CONTRM (100,1,
    └.1 FUNCM/(IM3_C
      └.1 TC LMISC1
```

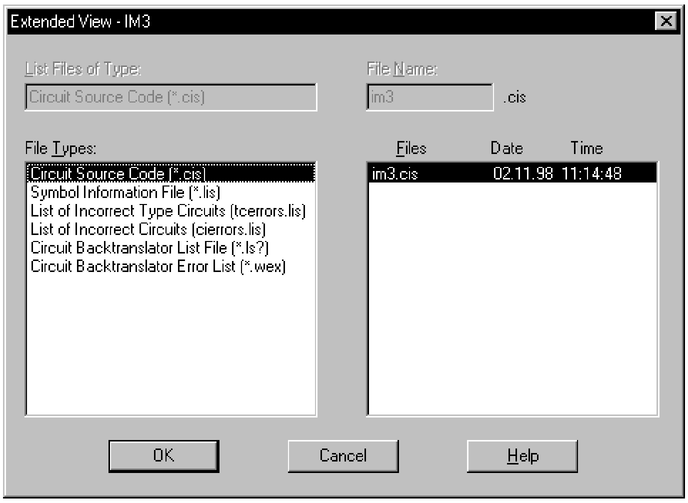
```
PC1.1.1/(IM3_Circuit IM3)
```

```
PC1.1.1/(IM3_Circuit IM3)
```

Figure 4-20. Open View Node Function Chart for Circuit Aspect

View Source Text

The **Extended View** dialog is shown with text / source code files on the selected aspect.



List Files of Type: Circuit Source Code (*.cis)

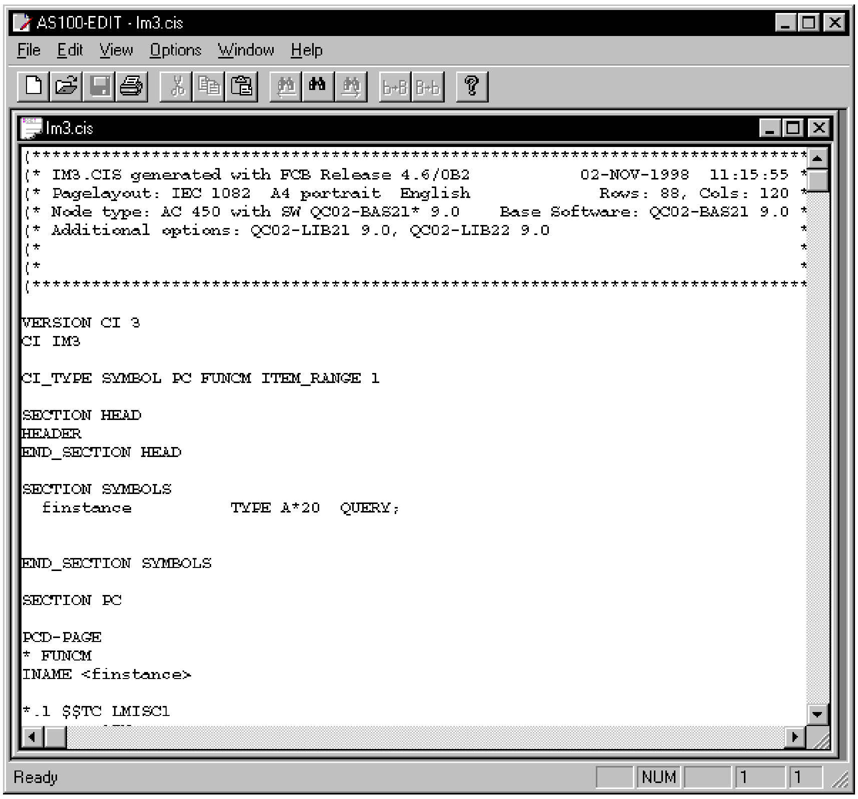
File Name: im3 .cis

File Types:
- Circuit Source Code (*.cis)
- Symbol Information File (*.lis)
- List of Incorrect Type Circuits (tcerrors.lis)
- List of Incorrect Circuits (cierrors.lis)
- Circuit Backtranslator List File (*.ls?)
- Circuit Backtranslator Error List (*.wex)

Files	Date	Time
im3.cis	02.11.98	11:14:48

Figure 4-21. Open View Source Text on Circuit Aspect/Extended View Dialog

You can start the text editor (AS100-Edit) on a selected source file.



```
(******************************************************************************
(* IM3.CIS generated with FCB Release 4.6/0B2          02-NOV-1998  11:15:55 *
(* Pagelayout: IEC 1082  A4 portrait  English             Rows: 88, Cols: 120 *
(* Node type: AC 450 with SW QC02-BAS21* 9.0    Base Software: QC02-BAS21 9.0 *
(* Additional options: QC02-LIB21 9.0, QC02-LIB22 9.0                         *
(*                                                                            *
(*                                                                            *
(******************************************************************************

VERSION CI 3
CI IM3

CI_TYPE SYMBOL PC FUNCM ITEM_RANGE 1

SECTION HEAD
HEADER
END_SECTION HEAD

SECTION SYMBOLS
  finstance          TYPE A*20  QUERY;


END_SECTION SYMBOLS

SECTION PC

PCD-PAGE
* FUNCM
INAME <finstance>

*.1 $$TC LMISC1
```

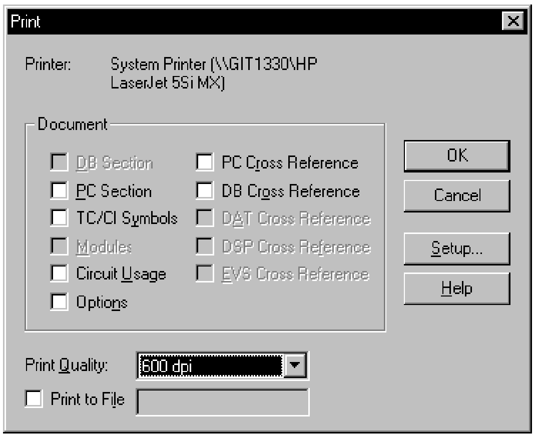
Figure 4-22. Open View Source Text on Circuit Aspect / AS100-EDIT

4.3.10 Print Aspect

You can print documents for the following aspects with the **Print** command of the command popup menu:

- Type Circuit / Circuit aspect for an Advant object representing a physical object
- DB Element aspect for an Advant object representing a board or signal or another kind of DB element
- DB Section aspect for an Advant object representing the boards and assigned signals or other kind of DB element of the node or part of a node.

Activating the **Print** command starts up the **Print** dialog of the type circuit or circuit involved.



Printer: System Printer (\\GIT1330\HP LaserJet 5Si MX)

Print Quality: 600 dpi

Figure 4-23. Print Circuit Aspect

All document generation functions can be performed on the type circuit or circuit as known from Function Chart Builder.

4.3.11 Delete Aspect

You can delete the following aspects with the **Delete** button of the **Object Window**:

- Type Circuit / Circuit aspect for an Advant object representing a physical object
- DB Element aspect for an Advant object representing a board or signal or another kind of DB element
- DB Section aspect for an Advant object representing the boards and assigned signals or other kind of DB element of the node or part of a node.

Pressing the **Delete** button by default deletes the aspect including the aspect data in the file system.

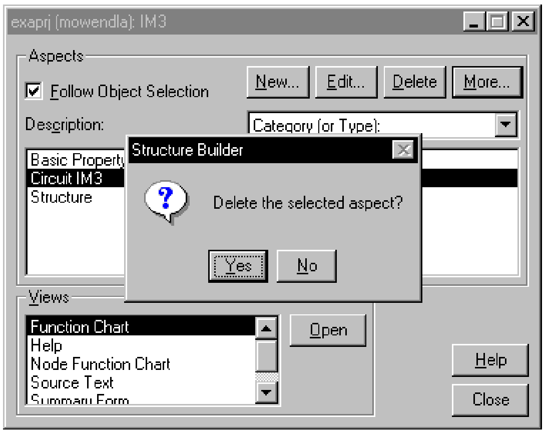
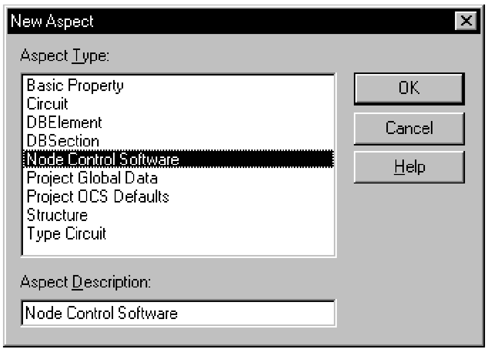


Figure 4-24. Delete Circuit Aspect

4.3.12 Create Node Control Software Aspect

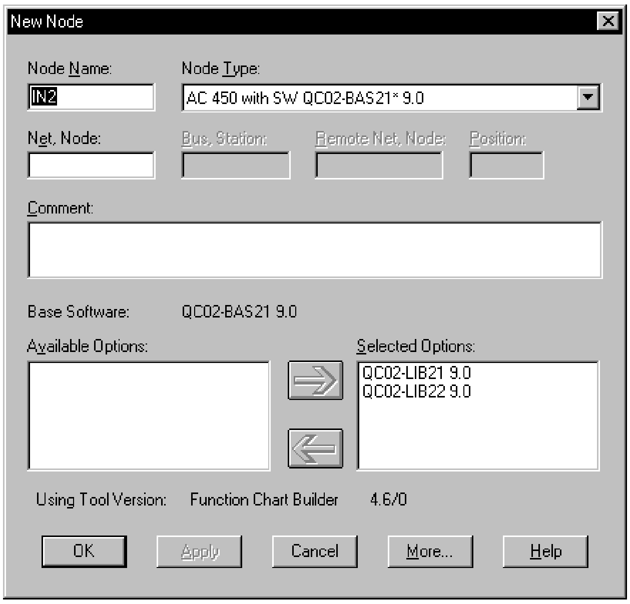
You can create a new Node Control Software aspect with the **New** button and **New Aspect** dialog of the **Object Window**.



Aspect Description: Node Control Software

Figure 4-25. New Node Control Software Aspect

Pressing the OK button in the **New** dialog shows the **New Node** dialog of Application Builder on the Node Control Software aspect to allow you to create a new node.

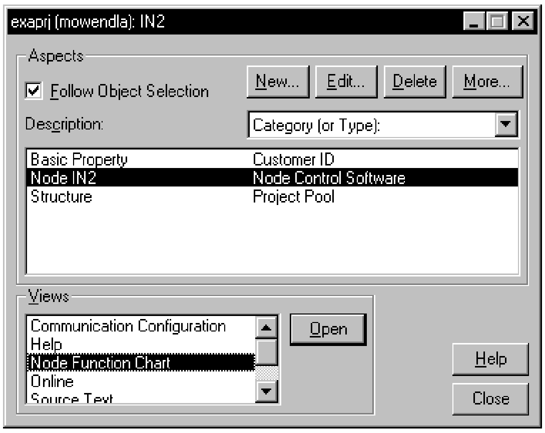


Node Name: IN2
Node Type: AC 450 with SW QC02-BAS21* 9.0
Base Software: QC02-BAS21 9.0
Selected Options:
QC02-LIB21 9.0
QC02-LIB22 9.0
Using Tool Version: Function Chart Builder 4.6/0

Figure 4-26. Add Node Control Software Aspect / New Node Dialog

The first eight characters of the aspect description (see **Object Window** Figure 4-3) are offered as node name. You can overwrite the proposed name of the new node and enter node type, addresses for net, node, bus, station, a comment and select options and tool versions.

After you have pressed the **OK** button in the **New Node** dialog the **Object Window** shows the added Node Control Software aspect.



☑ Follow Object Selection
Category (or Type):

Basic Property	Customer ID
Node IN2	Node Control Software
Structure	Project Pool

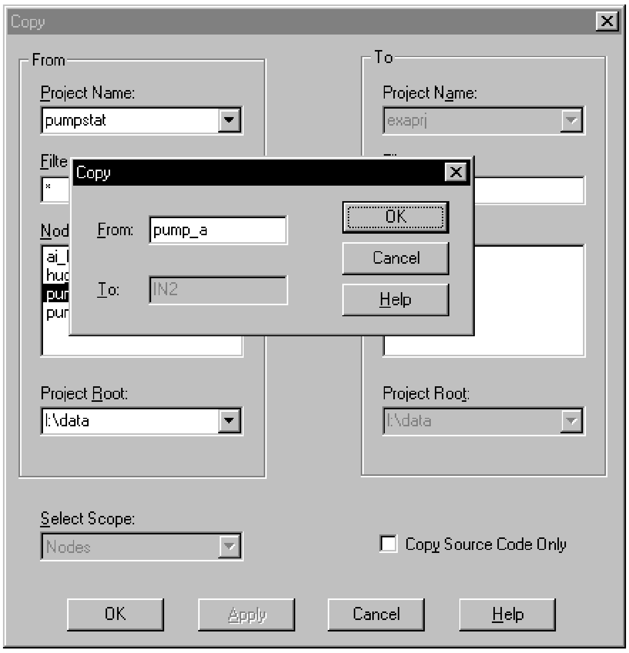
Communication Configuration
Help
Node Function Chart
Online
Source Text

Figure 4-27. Object Window with Added Node Control Software Aspect

4.3.13 Attach Node Control Software Aspect

You can attach the data of an existing node to a selected Node Control Software aspect by using the **Attach** command from the command popup menu of the aspect in the **Object Window**.

The **Attach** command shows the **Copy** dialog.



From

Project Name: pumpstat

Filter: *

Project Root: I:\data

To

Project Name: exaprj

Project Root: I:\data

Select Scope: Nodes

☐ Copy Source Code Only

Copy

From: pump_a

To: IN2

Figure 4-28. Copy Dialog for Attach Node Control Software Aspect

It is shown with scope fixed to Nodes, **To** section fixed to the current project and project root, **To** name in the name entry subdialog fixed to the current name. Before activating the **OK** button you can select in the **From** section a source project and project root and you have to select the source node.

Additionally you can select the **Copy Source Code Only** option.

After performing the **Copy** operation node data for the node are available as given by the source node.

4.3.14 Copy Node Control Software Aspect

You can create a Node Control Software aspect of an Advant object representing an OCS node with the **Edit | Copy** and **Edit | Paste (Special)** sequence of object commands when duplicating an object.

The **Copy/Paste** shows the **Copy** (Node) dialog of Application Builder.

It is shown with scope fixed to Nodes, **From** section fixed to current project, node name and project root. **To** section fixed to current project and project root. Before activating the **OK** button you can select the **Copy Source Code Only** option, after activating the OK button you have to enter the name of the new node for the new Node Control Software aspect in the Copy subdialog.

4.3.15 Open Node Control Software Aspect

You can open and edit the Node Control Software aspect with the **Open** button on a selected view. Another possibility to open the aspect is to display the aspect's command popup menu, activate the command's cascaded view menu and activate the appropriate view.

Open starts up the function to handle the view.

View Node Function Chart

Function Chart Builder started up in node mode is started up showing the application window.

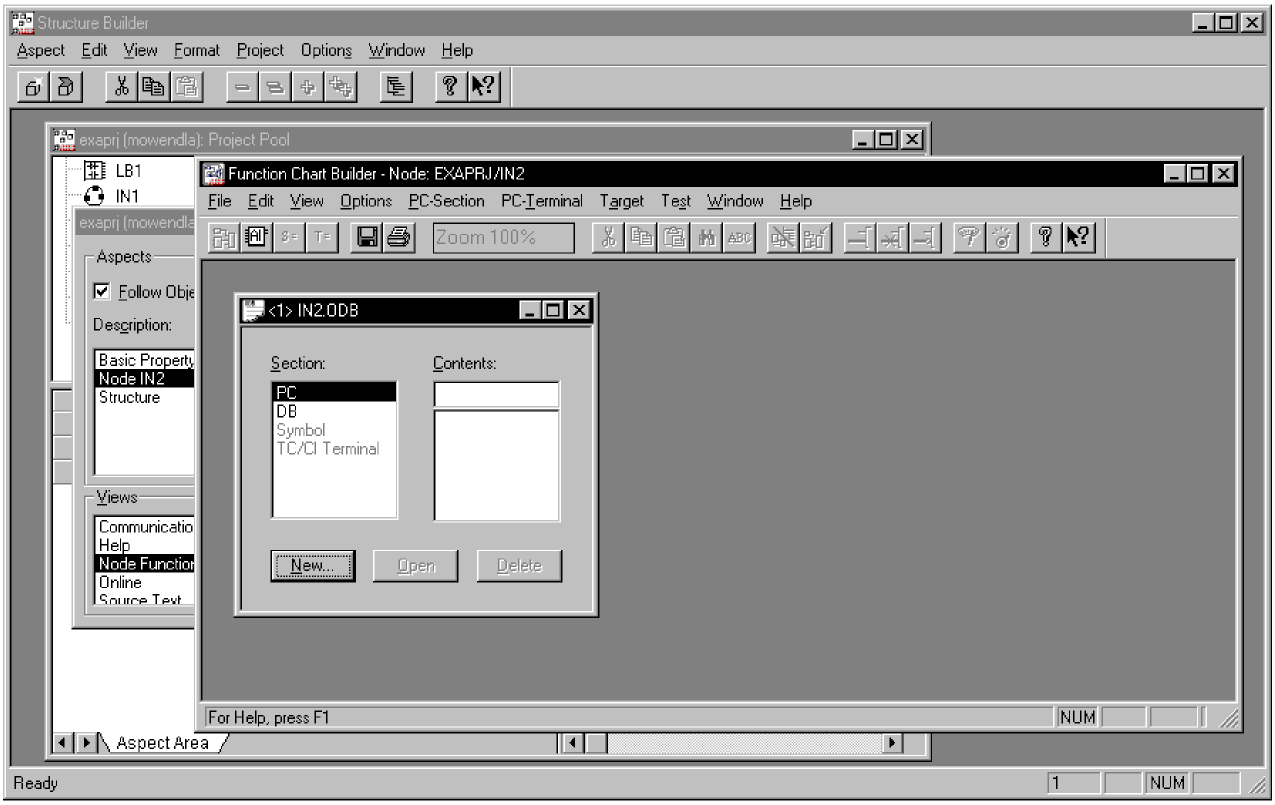


Figure 4-29. Open Node Control Software Aspect with Node Function Chart View

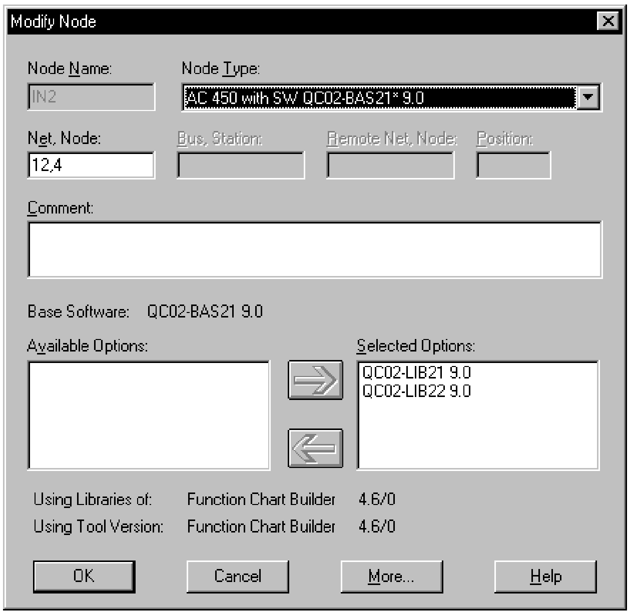
You can use editing and other Function Chart Builder functions within the application window on the node as known when working with the Function Chart Builder in independent mode. Detail views (MDI windows) shown in the application window initially depends on the selection done in the previous session on that aspect.

The initial view shown in the application window when Function Chart Builder is started up the first time, is the ODB window where detail view (section) selection can be done.

Other Functions (**Generate Source, Backtranslate, Check PC Source, Print, …, On-line** functions) of the Function Chart Builder are accessible via the **File** menu, **Target** menu, and **Test** menu of the Function Chart Builder application window.

View Summary Form

The **Modify Node** dialog of Application Builder is shown for the node represented by the Node Control Software aspect.



Node Name: IN2
Node Type: AC 450 with SW QC02-BAS21* 9.0
Net, Node: 12,4
Base Software: QC02-BAS21 9.0
Selected Options: QC02-LIB21 9.0
QC02-LIB22 9.0
Using Libraries of: Function Chart Builder 4.6/0
Using Tool Version: Function Chart Builder 4.6/0

Figure 4-30. Open Node Control Software Aspect with Summary Form View

View On-line

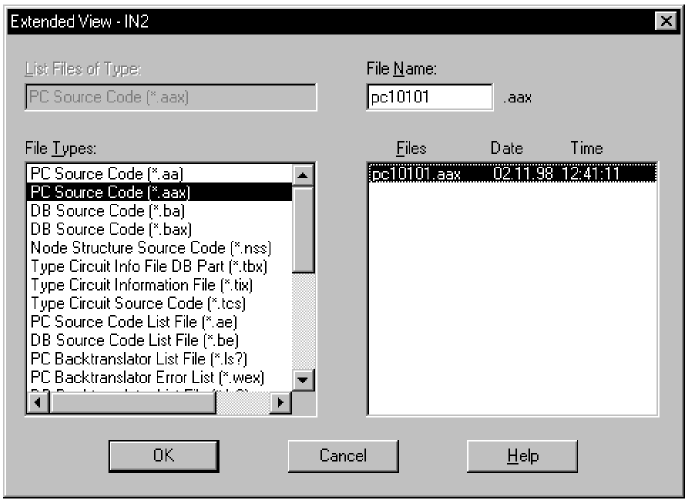
On-line Builder is started.

View Communication Configuration

Bus Configuration Builder is started.

View Source Text

The **Extended View** dialog (see Figure 4-31) is shown with text / source code files (binary files filtered out) on the selected Node Control Software aspect's node subdirectory tree.



List Files of Type: PC Source Code (*.aax)

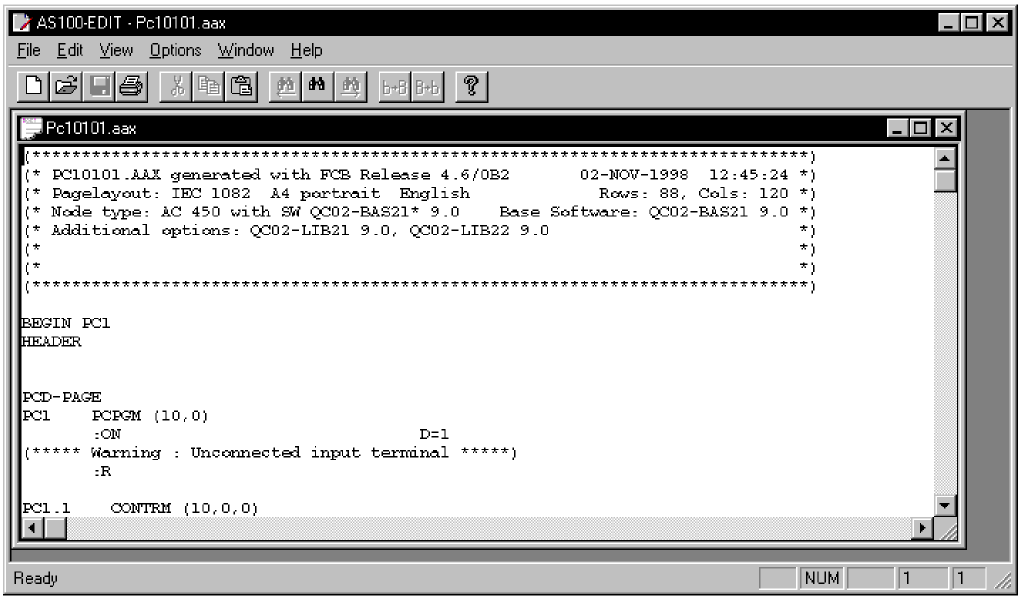
File Name: pc10101 .aax

File Types:
PC Source Code (*.aa)
PC Source Code (*.aax)
DB Source Code (*.ba)
DB Source Code (*.bax)
Node Structure Source Code (*.nss)
Type Circuit Info File DB Part (*.tbx)
Type Circuit Information File (*.tix)
Type Circuit Source Code (*.tcs)
PC Source Code List File (*.ae)
DB Source Code List File (*.be)
PC Backtranslator List File (*.ls?)
PC Backtranslator Error List (*.wex)

Files	Date	Time
pc10101.aax	02.11.98	12:41:11

Figure 4-31. Open Node Control Software Aspect with View Source Text / Extended View

You can start up text editor (AS100-Edit) on a selected source file.



```
(******************************************************************************)
(* PC10101.AAX generated with FCB Release 4.6/0B2       02-NOV-1998  12:45:24 *)
(* Pagelayout: IEC 1082  A4 portrait  English             Rows: 88, Cols: 120 *)
(* Node type: AC 450 with SW QC02-BAS21* 9.0    Base Software: QC02-BAS21 9.0 *)
(* Additional options: QC02-LIB21 9.0, QC02-LIB22 9.0                         *)
(*                                                                            *)
(*                                                                            *)
(******************************************************************************)

BEGIN PC1
HEADER


PCD-PAGE
PC1    PCPGM (10,0)
       :ON                            D=1
(***** Warning : Unconnected input terminal *****)
       :R

PC1.1    CONTRM (10,0,0)
```

Figure 4-32. Open Node Control Software Aspect with View Source Text / AS100-Edit

4.3.16 Print Node Control Software Aspect

You can print documents for a Node Control Software aspect with the **Print** command of the aspect’s command popup menu. Activating the **Print** command displays the command’s cascaded view menu. Activating the appropriate view starts the AMPL Control Configuration functions for printing the documents.

Print starts up Function Chart Builder’s **Print** dialog for the node.

View Node Documents

The **Print** dialog of Function Chart Builder is shown in node mode.

All document generation functions can be done on the node as known from Function Chart Builder.

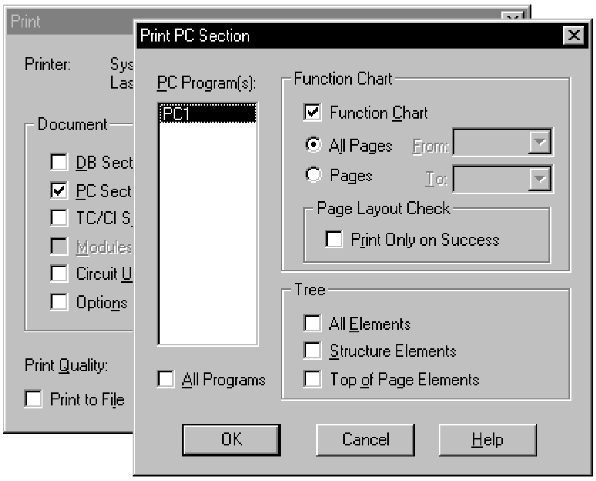
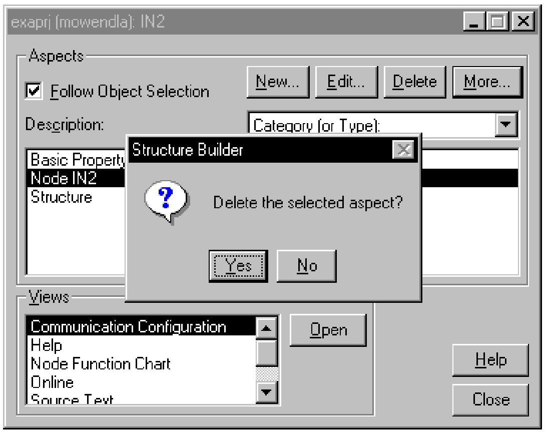


Figure 4-33. Print Node Control Software Aspect, Node Documents View

4.3.17 Delete Node Control Software Aspect

You can delete a Node Control Software aspect of a node Advant object with the **Delete** button of the **Object Window**.

By default, activating the **Delete** button leads to deletion of the selected Node Control Software aspect including the aspect data in the file system.



Delete the selected aspect?

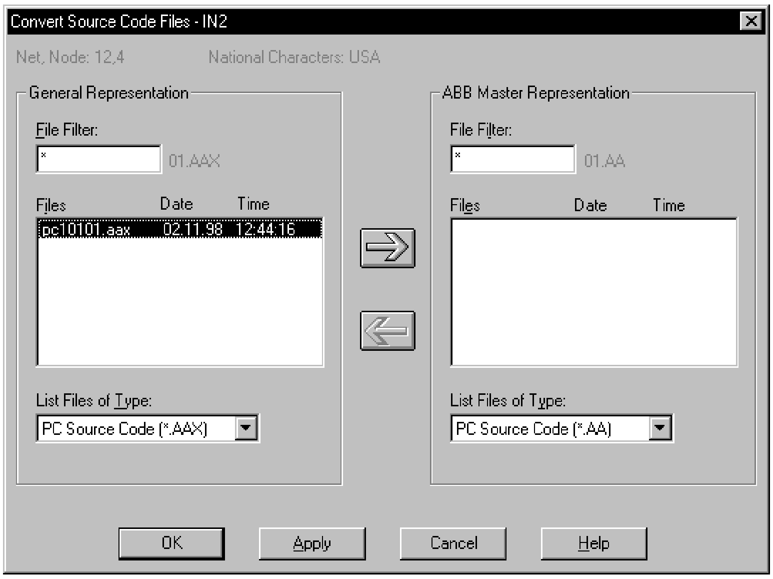
Figure 4-34. Delete Node Control Software Aspect

4.3.18 Convert Node Control Software Aspect

You can convert source code data of a Node Control Software aspect of a node Advant object with the **Convert** command from the aspect's command popup menu.

Use this command to convert source code generated with Function Chart Builder for download with On-line Builder and to convert source code uploaded with On-line Builder for use with Function Chart Builder.

The **Convert** command shows the **Convert Source Code Files** dialog of Control Configuration Functions for the Node represented by the selected Node Control Software aspect.



Convert Source Code Files - IN2

Net, Node: 12,4 National Characters: USA

General Representation

File Filter: * 01.AAX

Files	Date	Time
pc10101.aax	02.11.98	12:44:16

List Files of Type: PC Source Code (*.AAX)

ABB Master Representation

File Filter: * 01.AA

List Files of Type: PC Source Code (*.AA)

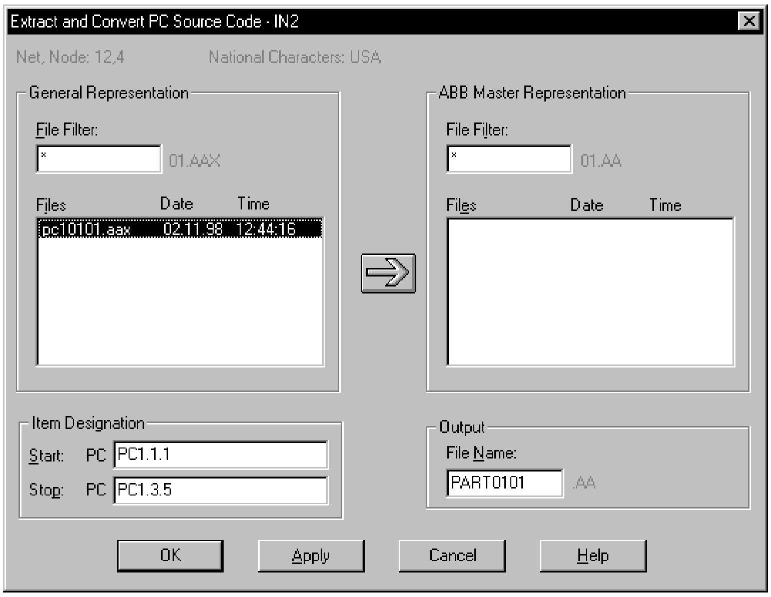
Figure 4-35. Convert Node Control Software Aspect

4.3.19 Extract Node Control Software Aspect

You can extract source code data of a Node Control Software aspect of a node Advant object with the **Extract** command from the aspect's command popup menu.

Use this command to extract certain parts from source code and convert it for download use with On-line Builder.

The **Extract** command shows the **Extract** and **Convert Source Code Files** dialog of Application Builder for the node represented by the selected Node Control Software aspect.



Extract and Convert PC Source Code - IN2

Net, Node: 12,4
National Characters: USA

General Representation
File Filter: * 01.AAX

Files	Date	Time
pc10101.aax	02.11.98	12:44:16

ABB Master Representation
File Filter: * 01.AA

Item Designation
Start: PC PC1.1.1
Stop: PC PC1.3.5

Output
File Name: PART0101 .AA

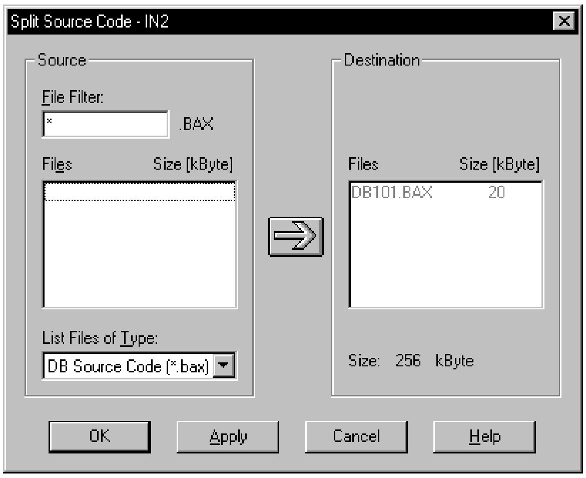
Figure 4-36. Extract Node Control Software Aspect

4.3.20 Split Node Control Software Aspect

You can split source code data of a Node Control Software aspect for use with On-line Builder with the **Split** command from the aspect's command popup menu.

Use this command to split DB or PC source code for download. This may be necessary if On-line Builder refuses to download the file because it is too big.

The **Split** command shows the **Split Source Code** dialog of Application Builder for the node represented by the selected Node Control Software aspect.



Split Source Code - IN2

Source

File Filter: * .BAX

Files Size [kByte]

List Files of Type: DB Source Code (*.bax)

Destination

Files Size [kByte]

DB101.BAX 20

Size: 256 kByte

Figure 4-37. Split Node Control Software Aspect

4.3.21 Concatenate

You can concatenate continuation DB or PC source code files (created by On-line Builder) to one single source code file with the **Concatenate** command. The Convert command concatenates such files implicitly. However, you may have existing continuation files that you want to concatenate.

4.3.22 Lock/Unlock

You can lock or unlock one or more Node Control Software, Circuit, Type Circuit, DB Element, or DB Section aspects. The lock mechanism of Application Builder is used. It has two different levels of locking:

- You can lock an object explicitly to prevent it from access by other users. Use the **Lock/Unlock** toggle command to lock/unlock an object. Each locked object must be unlocked manually. Use the explicit locking mechanism to prevent access also for those times, when no tool is running on that object.
- While you work on a node, circuit or type circuit with a tool, the aspect is locked automatically for you and all other users as in case of explicit locking. Though, if you have access to a node, you can run Function Chart Builder and On-line Builder on it in parallel. An aspect is unlocked automatically when you stop working with it.

4.3.23 Configuring Advant Objects for Library

4.3.23.1 Objects for Insertion of Process Control Functions via Drag and Drop

1. Configure the type circuit. Use allowed parameter references (see *Parameter Builder User´s Guide, Appendix on How to Define References)*.

   Make sure that the actual values of symbols are supplied either by default values or by parameter references. (Actual values supplied during instantiation into the circuit or by editing symbol values in the TC instances will get lost at drag and drop or **Locate Circuit** and execution of **Update Parameter**).

   For symbol controlled default connections of output terminals to DB element terminals make sure that the ‘=’ sign is outside of any symbol used in the default connection specification.

   You can easily adapt existing type circuits created and used for example with ACC in independent mode by substituting PU symbols with IU symbols and corresponding parameter references and adding parameter references to existing IU symbols.

   According to the parameter references parameter aspect types have to be defined using Parameter Builder´s Aspect Type Configuration and to be used for creation of the appropriate parameter aspects.

2. Configure the Circuit aspect.

   a. Introduce the circuit instance symbol `finstance`, data type A(72) is recommended.

   b. Begin the circuit of this aspect with a FUNCM (or another structure element) and give this element an instance name defined by the circuit's instance symbol `finstance`:
      `INAME <finstance>`.

   c. Instantiate the type circuit(s) into the circuit below the structure element.
      Make sure before you instantiate that the symbol substitution in Function Chart Builder circuit mode is switched off (uncheck **Options | Preferences | Substitute Symbols in Default Connections of TC/CI Terminals** in FCB)

Note that the TC instances do not need to get an instance name.

Instance names are given to the TC instances below the structure element automatically during drag and drop or **Locate Circuit**.

Note that the declaration length of the finstance symbol must fit to the length of its contents generated from basic property name of category Customer ID concatenated with _ (underscore) and the aspect description of the Circuit aspect.

Note that you can also configure a circuit without using a structure element. Then only one TC instance is allowed and this TC instance has to get the instance name <finstance>.

If the process object shall be used with the Locate Circuit command, then it must be placed in the OCS Structure below the node object it will be inserted into, and the relative reference designation entry in the OCS Structure aspect of the process object must contain the PC item designation of the insertion point PC element in the node.

4.3.23.2 Objects for Insertion of DB Elements via Drag and Drop

Besides drag and drop and **Locate Circuit** for objects with circuits that have a PC part, drag and drop and **Locate Circuit** of objects with circuits that have a DB part only is supported, too.

Parameter update for such circuits is supported if they have one DB type circuit instance that gets its instance name via a circuit symbol named `finstance` (`INAME <finstance>`). Allowed drop destination for DB circuits currently is a PC element in the tree or the function chart window of Function Chart Builder.

1. Configure the DB type circuit. Rules except the connection-related see above.
2. Configure the Circuit aspect
    a. Introduce the circuit instance symbol `finstance`, data type A(72) is recommended
    b. Instantiate the DB TC into the circuit using an instance name defined by the circuit's instance symbol `finstance`:
    `INAME <finstance>`.

NOTE

Only one DB TC instance is supported regarding parameter update.

4.3.23.3 Objects for Insertion of DB Elements via Generate DB

1. Configure the DB type circuit. Rules except the connection-related ones see above.
2. Configure the DB Element aspect by attaching the type circuit or/and editing it.

NOTE

Instance naming for the DB TC instance is done automatically when using the DB Element aspect with **Generate DB** command on a DB Section aspect.

4.3.23.4 PCPageBreak Parameter Aspect

Page break(s) of the contents of a circuit aspect of an Advant object inserted into a node in FCB can be modified according to three attributes of a parameter aspect of this object with predefined layout named PCPageBreak.

The three attributes are Page_Number, Headline and Control, data type for all the attributes can be Text(n).

Page_Number can provide the page number to be set and Headline the page comment to be set during insertion (by drag and drop or Locate Circuit) of the object into the node in FCB.

The control < specifies that TC instances after the first page break in the circuit having a page break also get the Headline from the first page break of the circuit. An empty control attribute means that these page break headlines are not affected by the integration functions.

Setting Page_Number and Headline follow the rules known from the **PC-Section | Page Layout | Modify Page Division** command of Function Chart Builder

To introduce a parameter aspect type named PCPageBreak with the three attributes use Parameter Builder **Edit | Aspect Type Configuration**.

After you have defined the parameter aspect type you can use this type to introduce an instance of PCPageBreak parameter for your Advant object in library with **Aspect | New** in Structure Builder.

4.3.23.5 Instance Names With/Without Absolute Reference Designation

You can configure the AMPL Control Configuration functionality to generate instance names for structure PC elements and for TC elements in FCB automatically on the base of the absolute reference designation in the Functional Structure or OCS structure by editing the <project>.ini file.

If the option AbsRefDesInstanceName in section [EOOptions] of the <project>.ini file is not set (AbsRefDesInstanceName=0 or no entry) the first part of the instance name (<firstpart>_<aspectdescription>[_<nr>]) is taken from the object name.

If the option AbsRefDesInstanceName=1 in section [EOOptions] of the <project>.ini file is set the first part of the instance name (<firstpart>_<aspectdescription>[_<nr>]) is taken from the absolute reference designation of the corresponding Advant object in the Function Structure or, if it does not exist from the absolute reference designation in the OCS Structure or, if it does not exist from the object name.

NOTE

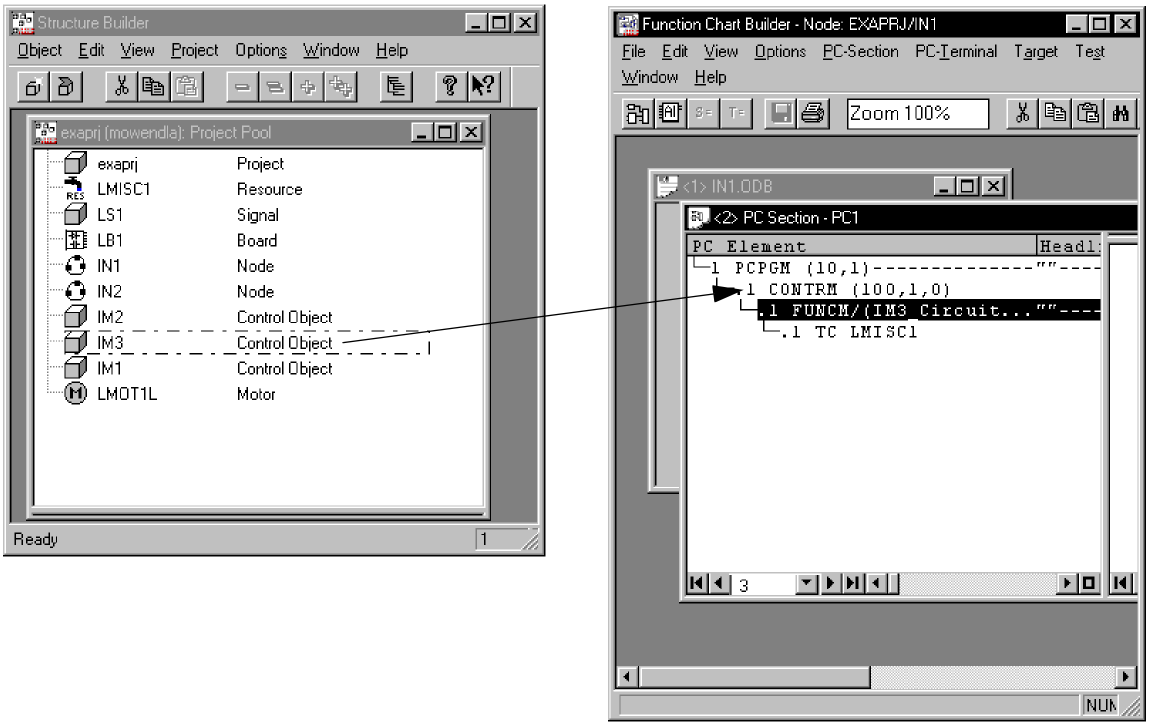
Up to (including) version 1.6/1 of ACC the first part of the instance name could only be taken from the object name.

4.3.24 Drag and Drop (Insert)

Insert

To locate the Circuit aspect's circuit in a PC program of a node in node mode and expand it there, use drag and drop or the Locate Circuit command (see Section 4.3.25, Locate Circuit).

The object to drag and drop is the Advant object having one or more Circuit aspects. It can be dropped on to PC elements shown in a PC program function chart and tree window.


Structure Builder
Object Edit View Project Options Window Help
exaprj (mowendla): Project Pool
Ready
1
Function Chart Builder - Node: EXAPRJ/IN1
File Edit View Options PC-Section PC-Terminal Target Test
Window Help
Zoom 100%
<1> IN1.ODB
<2> PC Section - PC1
PC Element
Headl:
3
NUM

exaprj	Project
LMISC1	Resource
LS1	Signal
LB1	Board
IN1	Node
IN2	Node
IM2	Control Object
IM3	Control Object
IM1	Control Object
LMOT1L	Motor

```
└1 PCPGM (10,1)--------------""----
  └1 CONTRM (100,1,0)
   └.1 FUNCM/(IM3_Circuit...""----
     └.1 TC LMISC1
```

Figure 4-38. Drag and Drop

Dropped circuits can participate in a circuit usage handling and parameter reference support specific for the Advant object environment. Circuit usage handling and insertion on same or lower level can be controlled by modifier keys pressed/not pressed during drag and drop.

Insertion of an object having a Circuit aspect into node mode function chart via drag and drop requires the following:

- The object has to be shown in some structure or pool window.
- Function Chart Builder has to be started up on the Node Control Software aspect of a node object with view Node Function Chart.
- The function chart or tree representation of the PC program of the node has to be shown, into which the object/Circuit aspect/circuit should be inserted.

Then the object can be selected in Structure Builder window with the mouse and dragged and dropped onto the PC element in the Function Chart Builder's function chart or tree representation window of the PC program in the node, where the circuit of the object's Circuit aspect shall be inserted.

Not pressing the **Shift** modifier key during drag and drop means that the circuit shall be inserted on lower level, pressing the **Shift** modifier key during drag and drop means that the circuit shall be inserted on same level.

During the drop action the circuit is located and expanded at the drop location in the PC program of the node and the parameter references are updated:

- The location is given by a next free item designation evaluated basing on the item designation of the selected element and the **Shift** key pressed or not pressed.
- The circuit contents is inserted, circuit borders are removed.
- The starting element (typically FUNCM) of the circuit gets the drop item designation and an instance name composed of the object's basic property name of category Customer ID concatenated with the Circuit aspect's aspect description.
- The parameter references of the instantiated TCs in the circuit are updated.
- The circuit is introduced into the usage administration of the AMPL Control Configuration integration functions depending on the **Ctrl** modifier key not pressed or pressed during drag and drop:
  - **Ctrl** modifier key not pressed: Circuit usage is to be administrated. (Only one insertion with **Ctrl** not pressed is allowed.)
  - **Ctrl** modifier key pressed: Circuit usage is not to be administrated.

If during drag and drop the **Ctrl** modifier key is pressed

- The circuit does not participate in the usage administration.
- No instance names are used.
- The parameters of the instantiated TCs are not updated.

This supports using the circuit of an object with Circuit aspect as some well tested code template as often as you like.

Instance names used in the circuit have to be unique in the scope of the PC program where the circuit shall be inserted.

If the circuit cannot be inserted due to expansion or instantiation problems, error message boxes will appear. To investigate the problem, follow the hint given in the message.

Connections that cannot be closed during drag and drop are introduced as virtual connections that can be consolidated to real connections later.

To be able to drag and drop an object with more than one Circuit aspect successfully, the aspect descriptions of the different Circuit aspects have to be different. This results in unique instance names for the starting structure (FUNCM) elements of the different circuits of the object. Unique instance names are one prerequisite to be able to insert the different circuits of the object into one PC program of a node.

If the starting structure (FUNCM) element instance names of circuits are not unique in the scope of the PC program then this is recognized during insertion and the insertion is refused by Function Chart Builder.

The item designation of the FUNCMs of the second and further circuits are derived from the item designation of the first circuit by simple incrementation. The first circuit's FUNCM item designation is the one determined by the place where the object is dropped.

If incrementation leads to a not allowed item designation this is recognized during insertion of this circuit, and the insertion is refused by Function Chart Builder.

Moving an already dropped circuit within the PC program and between PC programs using drag and drop is not supported. Within a PC program a move can be done by applying Function Chart Builder's functions **PC Section | Execution Order | Move Up** or **Move Down** on the starting structure element of the circuit (FUNCM).

Undo result of drop circuit is not supported via drag and drop and not within Function Chart Builder. You can use the **Initialize** command on the Circuit aspect with view Node Function Chart.

Initialize (Undo Insert)

The **Initialize** command, executed on a Circuit aspect that has been dropped into a PC program of a node successfully, works as follows:

1. The starting FUNCM and its subtree of elements in the PC program of the node is deleted.
2. The circuit usage administration is reset.
3. The Circuit aspect including the circuit file is kept.

Update Parameter

The **Update Parameter** command can be used for two different purposes:

- To prepare the updated source code (**Update Parameter** with view Function Chart)
- To update parameters of a circuit that has already be dragged and dropped into a PC program of a node (**Update Parameter** with view Node Function Chart).

After (initial) import of the consumer list or manually performed copy from library the consumer objects are available but the parameters are not yet updated.

Drag and drop (insert) the consumer objects (circuits) one by one into the PC programs in the nodes in node mode. (To make sure that all parameters are updated, this includes **Update Parameter** on Circuit aspect with view Node Function Chart.)

After re-import of the consumer list, which has changed parameter values, or after you manually edited parameter values, you have to update the parameters by calling **Update Parameter** command with view Node Function Chart. You call it either on Node Control Software aspect or Circuit aspect dependent on what has changed.

After re-import of the consumer list, which has introduced new objects, or after you manually copied new objects from library, you have to drag and drop (insert) the new consumer objects (circuits) one by one into the PC programs in the nodes in node mode. (To make sure that all parameters are updated, drop includes **Update Parameter** on Circuit aspect with view Node Function Chart.)

If you have both introduced new objects and changed parameter values for existing objects, you have to combine the corresponding working procedures.

4.3.25 Locate Circuit

General

The **Locate Circuit** command works similar as the drag and drop (insert) circuit function. Using the **Locate Circuit** command you do not need to open the Function Chart Builder on the Node Control Software aspect containing the PC program you want to insert the circuit.

The **Locate Circuit** command identifies where to locate the circuit in the PC program via the reference designation of the object placed in the OCS structure. This implies that

- The objects have been placed in the OCS structure
- Relative reference designations have been set to the PC item designation where the corresponding circuit shall be inserted.

This method supports you to define the location of a circuit in the PC program by a list entry, for example when process objects are imported. Then it is possible to locate the objects automatically in the OCS structure, based on the value of the relative reference designation attribute of the OCS structure aspect in the import data file.

You may use the **Locate Circuit** command for single or multiple objects. The command is available for the Circuit and Node Control Software aspect.

On Circuit Aspect

You may use the **Locate Circuit** command on a Circuit aspect to insert the contents of the circuit file into the PC program of the node, specified by the relative reference designation of the OCS structure aspect of the object that contains the Circuit aspect.

The node object is determined as the object in the OCS structure having a Node Control Software aspect and placed above the object carrying the Circuit aspect in the OCS structure.

The relative reference designation entry of the OCS structure aspect of the object containing the Circuit aspect, has to contain the PC item designation of the element in the PC program where you want to insert the circuit.

By default, the circuit is located on lower level with instance name and circuit usage support. To insert the circuit on same level, extend the PC item designation by the character “S”. With the character “L” you may define explicitly that the circuit shall be inserted on lower level. Some examples are defined in Table 4-4.

If the circuit is already inserted, the **Locate Circuit** command does not remove the circuit contents from the node, but displays a message that the circuit has already been inserted. To remove the circuit contents from the node, use the **Initialize** command, see Section 4.3.24, Drag and Drop (Insert).

To insert the circuit in a copy manner, without instance name and circuit usage support, extend the item designation by the character “C”, see Table 4-4 for examples.

If the circuit you want to insert is of kind DB circuit (type circuit instantiated in the circuit only creates or modifies DB elements), the relative reference designation entry is ignored.

Table 4-4. Examples of Relative Reference Designations for Locate Circuit Command

Relative Reference Designation	Circuit will be Inserted on	Instance Name/Circuit Usage Support
PC5.1	Lower level	Yes
PC5.1L	Lower level	Yes
PC5.1S	Same level	Yes
PC5.1C	Lower level	No
PC5.1LC	Lower level	No
PC5.1SC	Same level	No

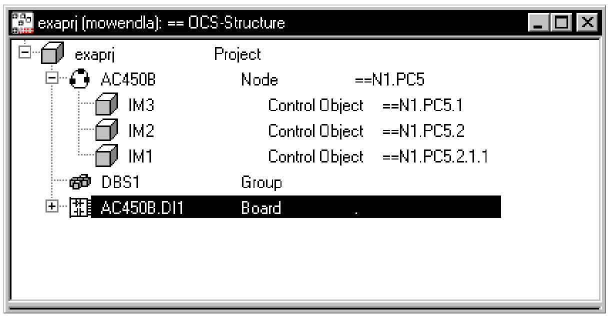
On Node Control Software Aspect

You may use the **Locate Circuit** command on a Node Control Software aspect to execute the **Locate Circuit** command on all Circuit aspects of objects placed in the OCS structure below the object having the Node Control Software aspects. For this command the same rules apply as for the command on the Circuit aspect.

To remove the contents of all circuits inserted with **Locate Circuit** command on the Node Control Software aspect from the node, you can use the **Initialize** command on the Node Control Software aspect.

OCS Structure Example

Figure 4-39 shows an example of an OCS structure with reference designations containing the PC item designation, that are used by the Locate Circuit command.



```
exaprj          Project
  AC450B          Node            ==N1.PC5
    IM3             Control Object  ==N1.PC5.1
    IM2             Control Object  ==N1.PC5.2
    IM1             Control Object  ==N1.PC5.2.1.1
  DBS1            Group
  AC450B.DI1      Board           .
```

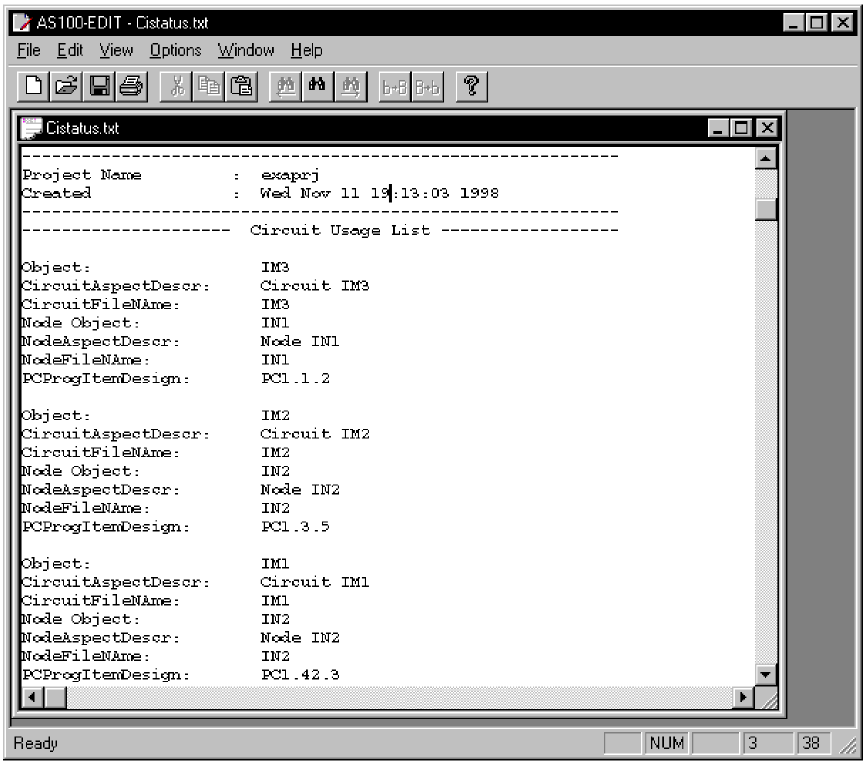
Figure 4-39. Example of OCS Structure with PC Item Designation

4.3.26 Circuit Usage List

The **Open** command with view **Circuit Usage List**, which is available for the Project OCS Defaults aspect, is useful if you have a project with many objects having Circuit aspects. You may want to know

- If the objects (the circuits of the circuit aspects) are already inserted (via drag and drop or via the Locate Circuit command) into the PC programs of the nodes, and
- Where they are located, for example in which node and PC program.

The **Circuit Usage List** view gives you a complete overview of all circuits in the project and their usage information. This information is written to a file named CiStatus.txt. This file is opened with the AS100-Edit.



```
-------------------------------------------------------------
Project Name         :  exaprj
Created              :  Wed Nov 11 19:13:03 1998
-------------------------------------------------------------
---------------------  Circuit Usage List ------------------

Object:                 IM3
CircuitAspectDescr:     Circuit IM3
CircuitFileNAme:        IM3
Node Object:            IN1
NodeAspectDescr:        Node IN1
NodeFileNAme:           IN1
PCProgItemDesign:       PC1.1.2

Object:                 IM2
CircuitAspectDescr:     Circuit IM2
CircuitFileNAme:        IM2
Node Object:            IN2
NodeAspectDescr:        Node IN2
NodeFileNAme:           IN2
PCProgItemDesign:       PC1.3.5

Object:                 IM1
CircuitAspectDescr:     Circuit IM1
CircuitFileNAme:        IM1
Node Object:            IN2
NodeAspectDescr:        Node IN2
NodeFileNAme:           IN2
PCProgItemDesign:       PC1.42.3
```

Figure 4-40. Circuit Usage List File CiStatus.txt Opened with AS100-Edit

The usage information consists of:

- Object name (basic property name of category Customer ID)
- Circuit aspect description
- Circuit file name

If the circuit is located into the PC program of a node, the following information is also available:

- Node object name (basic property name of category Customer ID)
- Node Control Software aspect description
- Node file name
- PC program item designation (drop or insert location)
- Instance name of the located circuit

Object names and aspect description each can have up to 50 characters. The list can be printed with standard page layout information supported by AS100-Edit.

4.3.27 Parameter References

4.3.27.1 Configure Parameter References

You can configure parameter references for consumer type circuits (used in circuits represented by Circuit aspects of consumer Advant objects) and for board or signal or other type of DB element type circuits.

For configuration of the parameter references you can use either the Function Chart Builder symbol editor (if the type circuit has no variants and/or loops) or a source code editor.

The basic way to configure a reference is:

- Define a symbol for the item to be parameterized, for example,

  ```
  IO_UNIT      TYPE A*5   QUERY := “DX100”;
  ```

- Define a companion symbol to specify the reference for the original symbol in the (default) value part of the companion symbol, for example,

  ```
  IO_UNIT_REF  TYPE A*40  QUERY :=
                 “$$REF.ProcessObjectParameter.IO_UNIT”;
  ```

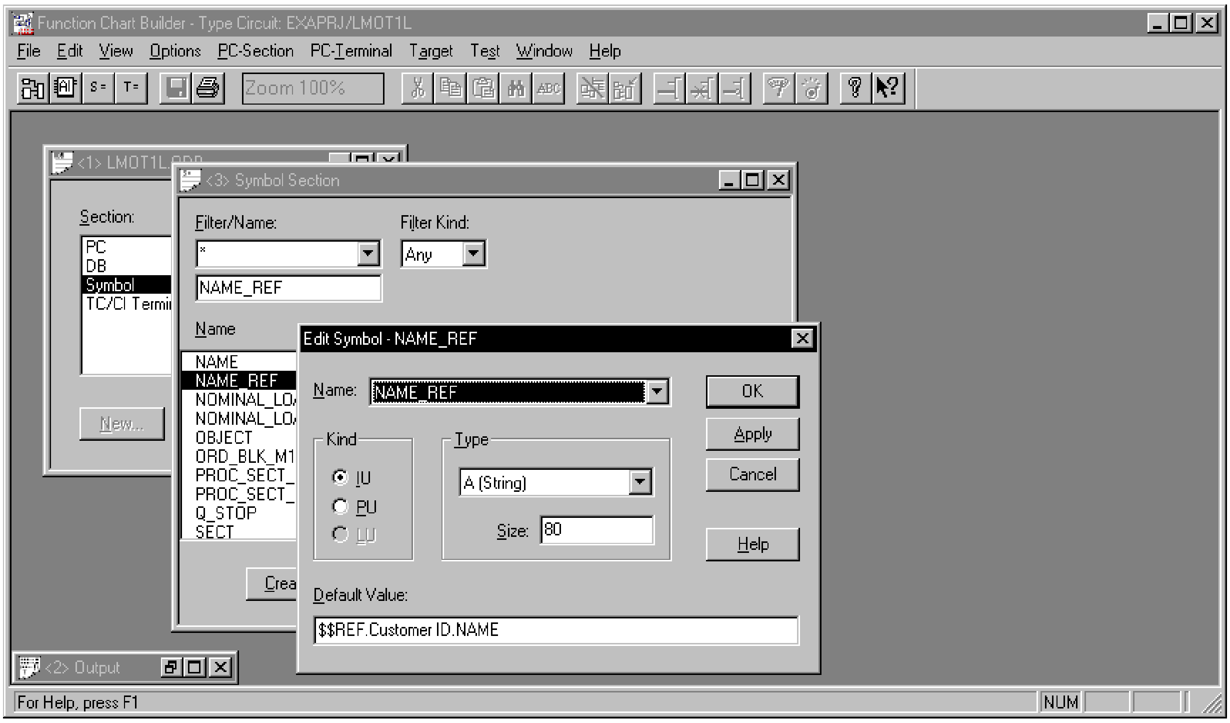
This is a relative reference to the IO_UNIT attribute of the ProcessObjectParameter aspect of the same object.

In principle the reference specification can be of one of three different types

- Relative reference to the same object
- Relative reference to another (related) object
- Absolute reference to any object

and has to conform to the syntax defined in *Parameter Builder User´s Guide, Appendix on How to Define References*.

The examples in Figure 4-41 and Figure 4-42 show relative references to the parameter aspect of the own (same) Advant object.



Edit Symbol - NAME_REF

Filter/Name: *

Filter Kind: Any

NAME_REF

Name: NAME_REF

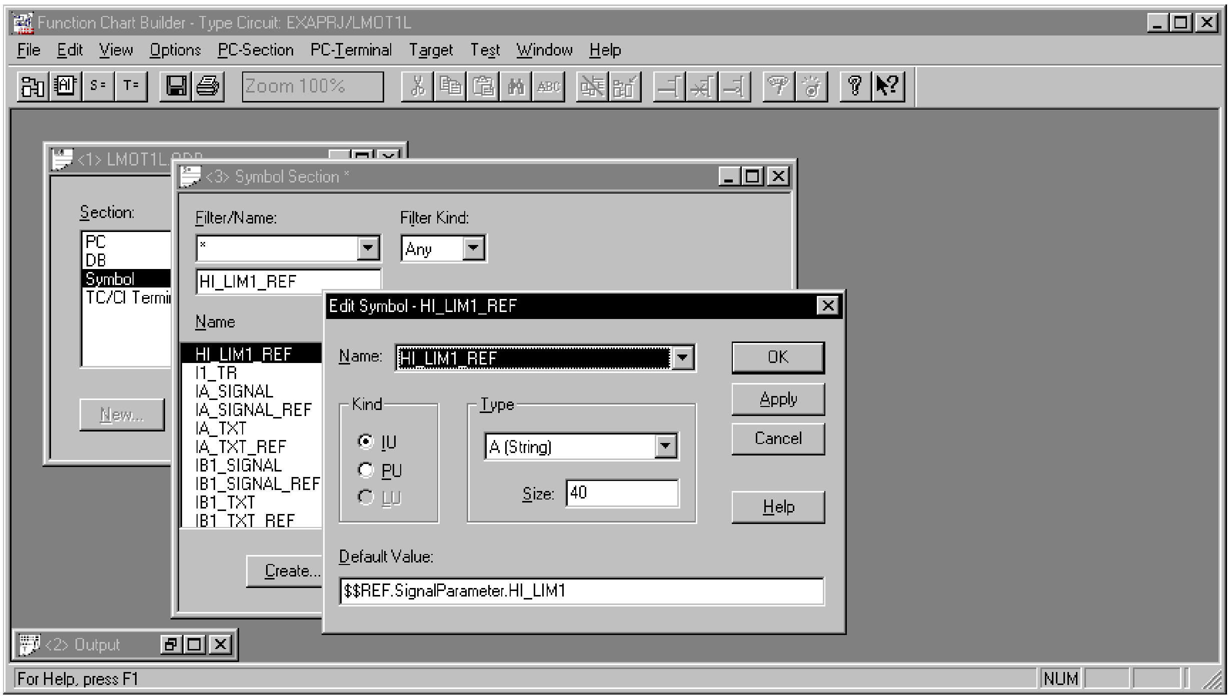
Kind: ● IU ○ PU ○ LU

Type: A (String)

Size: 80

Default Value: $$REF.Customer ID.NAME

Figure 4-41. Reference Configuration of TC for Circuit Aspect of a Consumer Advant Object (Example)



Edit Symbol - HI_LIM1_REF

Filter/Name: *

Filter Kind: Any

HI_LIM1_REF

Name: HI_LIM1_REF

Kind: ● IU ○ PU ○ LU

Type: A (String)

Size: 40

Default Value: $$REF.SignalParameter.HI_LIM1

Figure 4-42. Reference Configuration of Signal TC for DB Element Aspect of a Signal Advant Object (Example)

Parameter references are executed for the following aspects:

- Circuit aspects of consumer Advant objects by **Update Parameter** commands on the following aspects:
  - Project OCS Defaults aspect of a object representing a project object
  - Node Control Software aspect representing a node
  - The individual Circuit aspect of an object representing a consumer (such as a motor) (see Section 4.3.27.2, Update Parameter)
- DB element aspects of board/signal Advant object by the **Generate DB** command on the DB Section aspect of an Advant object representing all or a certain amount of the boards and signals of a node (see Section 4.3.28.1, Configure DB Element Aspect for Board Advant Object and Section 4.3.28.5, Generate DB).

You have to use these commands explicitly when you want the new parameter values to be taken from the parameter aspects.

Remarks

- Parameter referencing is handled by copying the parameter values from the Parameter Builder to the internal data repository of the Control Configuration Functions.
- In most cases symbol name of a consumer type circuit or board-/signal type circuit equals a attribute (column) name of the parameter aspect provided by the Parameter Builder. For cases where this cannot be applied a mapping between symbol name and parameter name is supported through the reference specification.
- The parameter values entered into the parameter aspect have to follow the exact AMPL source code value syntax. No conversion of parameter values to symbol values is provided.
- The consumer TC symbol interface and the board / signal / DB element TC symbol interface (the parameter set) has to be fixed for a project.
- When a TC to be used in a project changes it’s symbol interface the corresponding library Advant objects have to be adapted as follows:
  a. Remove the old type circuit instances from the circuit.
  b. Instantiate the new type circuit in the circuit (and possibly reconnect to other elements in the circuit)
  c. Correct the corresponding parameter aspect / the parameter aspect table (for example, by including a new parameter column for a new symbol)
  d. After that the changed library Advant object can be used.

  Old instances (instances created from the old library object) are not handled by the change procedure, they have to be removed from the project and recreated again in your responsibility.
- Parameter referencing is generally not supported for symbols in consumer circuits (circuits holding consumer TC instances). An exception is the `finstance` symbol for circuits that are to be dropped into Function Chart Builder node mode.

4.3.27.2 Update Parameter

The **Update Parameter** command is registered by the AMPL Control Configuration integration functions for the following aspect types:

- Project OCS Defaults aspect type (for an object representing a project)
- Circuit aspect type (for an object representing a consumer object)
- Node Control Software aspect type
- DB Element aspect type
- DB Section aspect type.

The parameter reference update refers to providing the actual parameter value as the symbol value for the symbol that has the companion reference symbol which specifies the parameter reference within the type circuit instance. Parameter/symbol value entry errors are detected during update.

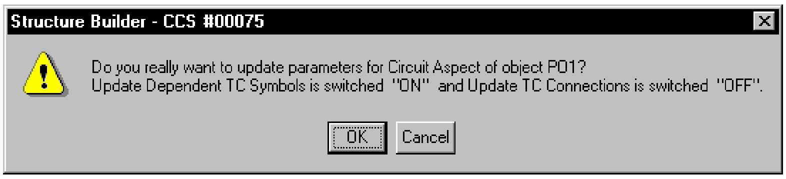
The symbol value change effect of the **Update Parameter** command is known from the Function Chart Builder function **Edit | Symbol Values**.

Update Parameter includes consolidation of virtual connections.

On Circuit Aspect

When the command is called from the **Object Window**'s command popup menu of the selected object, it updates the symbol values of all symbols with parameter references of the TC instances in the circuit of the Circuit aspect. Glue logic or other elements that had been inserted manually into the circuit will not be lost after update.

By default symbol update in **Update Parameter** includes update of dependent symbols and update of TC connections. This fact is stated in a warning box as changes done manually are overwritten.



Structure Builder - CCS #00075

Do you really want to update parameters for Circuit Aspect of object P01?
Update Dependent TC Symbols is switched "ON" and Update TC Connections is switched "OFF".

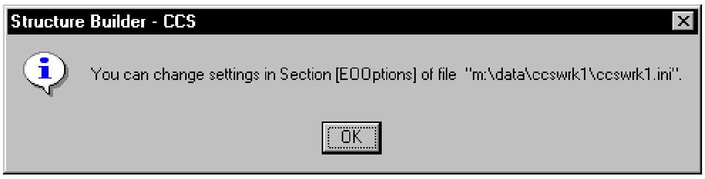
Figure 4-43. TC Symbol Update Switch Settings (Warning)

- Update dependent symbols is achieved by switching the TC instance in Function Chart Builder to default symbol values before changing (editing) the symbol values.
  Therefore changing symbol values on circuit level for symbols without parameter references before drag and drop has no effect.
  Further symbol values of symbols without parameter references edited manually in the TC instance are reset to the default values.
- Update TC connections overwrites manually changed TC connections of the TC instance.

NOTE

> Starting with Function Chart Builder version 4.7 the Update TC connections option needs no longer be switched off to prevent overwriting of manually changed TC connections. By default Function Chart Builder now keeps modified (manually changed) TC connections.

The warning box is displayed before the update. If you press the **Cancel** button, you get the hint that you can change the settings by editing the <project>.INI file.



You can change settings in Section [EOOptions] of file "m:\data\ccswrk1\ccswrk1.ini".

Figure 4-44. Change Settings Information

To switch the features off, set the corresponding option in the `EOOptions` section to 1, for example,

```
[EOOptions]
...
NotUpdateDependentSymbols=1
NotUpdateTCConnections=1
```

Update of TC connections (if controlled by symbols) and update of dependent symbols depend on the corresponding option settings. By default they are updated. Note that the settings are valid for the total project.

View Function Chart: Update Parameter updates the symbol values of symbols with parameter references of the TC instances in the circuit of the Circuit aspect and generates updated circuit source code.

View Node Function Chart: Update Parameter updates the symbol values of symbols with parameter references of the TC instances in the inserted circuit in the node of the Node Control Software aspect.

On Node Control Software Aspect

When the command is called from the **Object Window**’s command popup menu of the selected object, it updates the symbol values of all symbols with parameter references of the TC instances in all circuits inserted into the node of the Node object carrying the Node Control Software aspect.

Update Parameter chains this command on all circuit aspects with view Node Function chart.

On Project OCS Defaults Aspect

When the command is called from the **Object Window**'s command popup menu of the selected object, it updates the symbol values of all symbols with parameter references of the TC instances in all circuits of Circuit aspects in the project.

View Function Chart: Update Parameter updates the symbol values of symbols with parameter references of the TC instances in the circuits of all Circuit aspects in the project and generates updated circuit source code.

On DB Section Aspect

When the command is called from the **Object Window**'s command popup menu of the selected object, it updates the symbol values of all symbols with parameter references in the DB type circuit instances of the objects with DB Element aspects placed in the OCS Structure below the object holding the DB Section aspect.

View Function Chart: **Update Parameter** updates the symbol values of symbols with parameter references of the DB type circuit instances in the circuit of the DB Section aspect.

View Node Function Chart: **Update Parameter** updates the symbol values of symbols with parameter references of the DB type circuit instances in the node of the Node Control Software aspect of the Node object the DB Section aspect had been generated into.

On DB Element Aspect

When the command is called from the **Object Window**'s command popup menu of the selected object, it updates the symbol values of symbols of a set of TC instances in the scope that are configured with parameter references. The set is defined by the objects in the OCS substructure on the current level and below having the DB element aspect (for example, a board object and the signal objects allocated to it).

View Function Chart: **Update Parameter** updates the symbol values of symbols with parameter references of this set of DB type circuit instances in the circuit of the DB Section aspect.

View Node Function Chart: **Update Parameter** updates the symbol values of symbols with parameter references of this set of DB type circuit instances in the node of the Node Control Software aspect of the Node object the DB Section aspect had been generated into.

4.3.27.3 Generate DB

The **Generate DB** command is registered by the AMPL Control Configuration integration functions for the Project OCS Defaults aspect type (for an object representing a project), DB Element aspect type, and DB Section aspect type.

The effect of the command is that DB type circuit instances are inserted with updated parameters. If the DB type circuits already exist they are deleted before they are re-inserted. Parameter/symbol value entry errors are detected during generation.

On DB Section Aspect

When the command is called from the **Object Window**'s command popup menu of the selected object, it instantiates all DB type circuits of the objects with DB Element aspects placed in the OCS Structure below the object holding the DB Section aspect and it updates the symbol values of all symbols with parameter references in these DB type circuit instances

View Function Chart: Generate DB instantiates the DB type circuits into the circuit of the DB Section aspect and updates the symbol values of symbols with parameter references of the DB-TC instances

View Node Function Chart: Generate DB instantiates the DB type circuits with updated symbol values of symbols with parameter references into the node of the Node Control Software aspect of the Node object which is parent or ancestor of the object carrying the DB Section aspect.

On DB Element Aspect

When the command is called from the **Object Window**'s command popup menu of the selected object, it instantiates a set of DB type circuit instances in the scope and updates the symbol values of the symbols with parameter references of these TC instances. The set is defined by the objects in the OCS substructure on the current level and below having the DB element aspect (for example, a board object and the signal objects allocated to it).

View Node Function Chart: Generate DB instantiates and updates this set of DB type circuit instances in the node of the Node Control Software aspect of the Node object the DB Section aspect had been generated into.

On Project OCS Defaults Aspect

When the command is called from the **Object Window**'s command popup menu of the selected object, it instantiates a set of DB type circuit instances in the scope and updates the symbol values of the symbols with parameter references of these TC instances.

View Node Function Chart: Generate DB chains this command on all DB Sections aspects in the project.

4.3.27.4 Drag and Drop / Locate Circuit

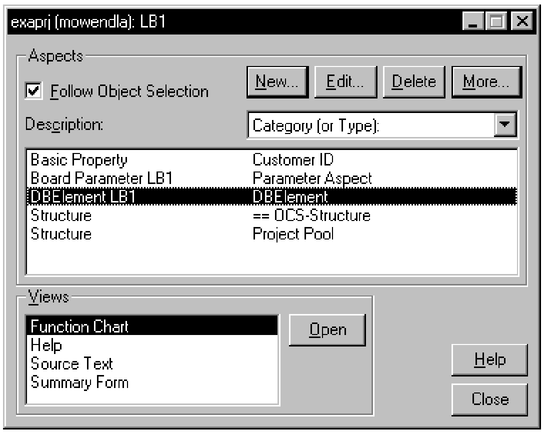
Drag and drop and the Locate Circuit command both include **Update Parameter** on the Circuit aspect with view Node Function Chart with the current option settings for Update TC connections and Update dependent symbols.

Further details see Section 4.3.24, Drag and Drop (Insert).

4.3.28 DB Element Handling

4.3.28.1 Configure DB Element Aspect for Board Advant Object

An Advant object representing a board has the following principle structure:



☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
Board Parameter LB1	Parameter Aspect
DBElement LB1	DBElement
Structure	== OCS-Structure
Structure	Project Pool

Function Chart
Help
Source Text
Summary Form

Figure 4-45. Aspects of an Advant Object Representing a Board

Example

Source code fragment for a board type circuit.

```
VERSION TC 3
TC AX52

TC_TYPE SYMBOL DB CREATE

SECTION HEAD
HEADER
END_SECTION HEAD


SECTION SYMBOLS
  binstance TYPE A*20 QUERY :="AX100";
  binstance_REF TYPE A*40 QUERY :="$$REF.== OCS-Structure.ABS";
END_SECTION SYMBOLS
SECTION DB
<binstance>         AX

<binstance>.1       AIXS
(*  Base part  *)
   :NAME            <binstance>.1
....
```

```
<binstance>.14      AIXS
(*  Base part  *)
   :NAME            <binstance>.14

<binstance>.15      AOXS
(*  Base part  *)
   :NAME            <binstance>.15
....
<binstance>.20      AOXS
(*  Base part  *)
   :NAME            <binstance>.20

END_SECTION DB
```

A board Advant object in library directly has a type circuit assigned.

A copy of a board Advant object from library in the project references the board type circuit of the corresponding object in library, there is no instance specific type circuit.

Within the project a board Advant object can be edited having an instance specific type circuit, too.

The relative reference to a structure aspect of the own Advant object included into the source code is handled when the data of the aspect of type DB Section (see DB section circuit in Section 4.3.28.5, Generate DB) is accumulated:

The absolute reference designation of the structure aspect with aspect description == OCS-Structure ($$REF.== OCS-Structure.ABS) within the control system oriented structure is acquired and the item designation part of it is used as value for the `binstance` symbol.

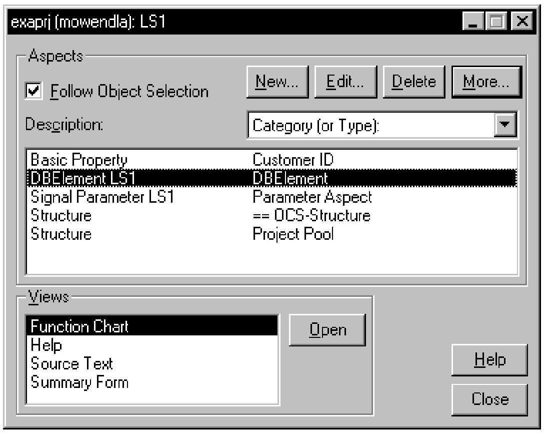
One DB Element aspect per board Advant object is supported.

NOTE

> The symbol that defines the board instance must be named `binstance` declared as in the example!

4.3.28.2 Configure DB Element Aspect for Signal Advant Object

An Advant object representing a signal has the following principle structure:



☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
DBElement LS1	DBElement
Signal Parameter LS1	Parameter Aspect
Structure	== OCS-Structure
Structure	Project Pool

Views:
- Function Chart
- Help
- Source Text
- Summary Form

Figure 4-46. Object Window of an Advant Object Representing a Signal

Example

Source code fragment of a signal type circuit

```
VERSION TC 3
TC AIXS
TC_TYPE SYMBOL DB MODIFY

SECTION HEAD
HEADER
END_SECTION HEAD

SECTION SYMBOLS
  rmax TYPE A*20  QUERY   := "100.000";
  rmin TYPE A*20  QUERY   := "-100.000";
  rmax_REF TYPE A*40 QUERY := "$$REF.SignalParameter.RANGEMAX";
  rmin_REF TYPE A*40 QUERY := "$$REF.SignalParameter.RANGEMIN";
  sname TYPE A*20  QUERY   := "nnnn";
  sinstance TYPE A*20  QUERY   := "AX100.1";
  sname_REF TYPE A*40  QUERY   := "$$REF.Customer ID.NAME";
  sinstance_REF TYPE A*40 QUERY :="$$REF.== OCS-Structure.ABS";
  description TYPE A*20  QUERY := "x"
  description_REF TYPE A*40 QUERY :=
                               "$$REF.Customer ID.DESCRIPTION";
...

END_SECTION SYMBOLS
```

```
SECTION DB

MODIFY     <sinstance>         AIXS
(*  Base part  *)
  :NAME            <sname>
  :DESCR           <description>
  :RANGEMAX        <rmax>
  :RANGEMIN        <rmin>
...

END_SECTION DB
```

A signal Advant object in library directly has a type circuit assigned.

A copy of a signal Advant object form library in the project pool references the board type circuit of the corresponding object in library, there is no instance specific type circuit.

Within the project a signal Advant object can be edited having an instance specific type circuit too.

The relative references to aspects of the own Advant object included into the source code are handled when the data of the aspect of type DB Section (see DB section circuit in Section 4.3.28.5, Generate DB) is accumulated:

- The absolute reference designation of the structure aspect with aspect description == OCS-Structure ($$REF.== OCS-Structure.ABS) within the control system oriented structure is acquired from Structure Builder and the item designation part of it is used as value for the `binstance` symbol.
- The name attribute of basic property aspect with aspect description Customer Identity ($$REF.Customer ID.NAME) is acquired from Structure Builder and used as value of the `sname` symbol (restricted to the allowed length and to the allowed characters)
- The RANGEMAX attribute of the parameter aspect with aspect description Signal Parameter ($$REF.Signal Parameter.RANGEMAX) is acquired from the Parameter Builder and used as the value of the symbol `rmax`.

One DB Element aspect per signal Advant object is supported.

NOTE

The symbol that defines the signal instance must be named `sinstance` and declared as in the example.

4.3.28.3 Configure DB Element Aspect for Other Kind of DB Element

A DB Element Advant object in library has a type circuit assigned.
In this type circuit the following references can be used:

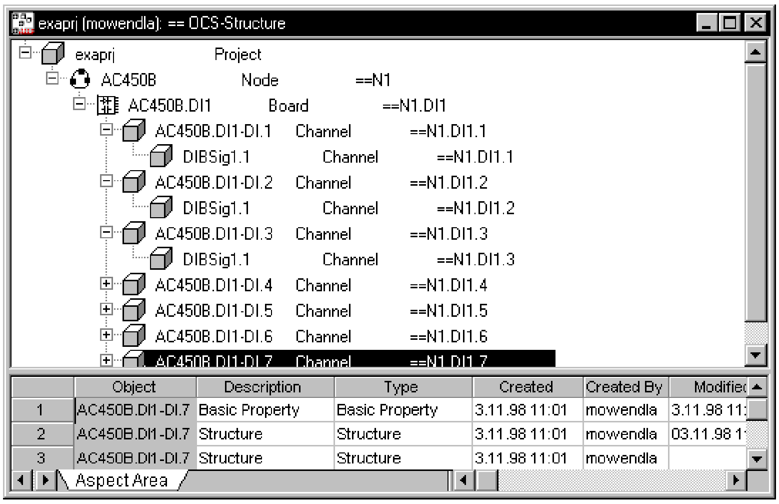
- An instance reference via the `binstance` symbol
- A name reference via the `sname` symbol
- A description reference via the `description` symbol
- Parameter references.

4.3.28.4 Building up OCS Structure

To plan the I/Os and to prepare for using the **Generate DB** command you have to build up the OCS structure in the following way:

1. Place the node Advant object in the OCS structure.
2. If you like to represent groups of your board Advant objects by special parent objects (further called board group objects) create these objects and place them as childs of the node object
3. Place the board objects together with their channel objects as childs of the board group objects. One approach is to have the needed board types and their channel objects grouped in substructures in the library and copy them into the OCS structure.
4. Place the signal objects as childs of the channel objects
5. Add one DB Section aspect to the node object or if you have board group objects add one DB Section aspect to each board group object

Then your OCS structure should look like the one in the following figure.



```
exaprj                  Project
  AC450B                    Node          ==N1
    AC450B.DI1                  Board         ==N1.DI1
      AC450B.DI1-DI.1               Channel       ==N1.DI1.1
        DIBSig1.1                       Channel       ==N1.DI1.1
      AC450B.DI1-DI.2               Channel       ==N1.DI1.2
        DIBSig1.1                       Channel       ==N1.DI1.2
      AC450B.DI1-DI.3               Channel       ==N1.DI1.3
        DIBSig1.1                       Channel       ==N1.DI1.3
      AC450B.DI1-DI.4               Channel       ==N1.DI1.4
      AC450B.DI1-DI.5               Channel       ==N1.DI1.5
      AC450B.DI1-DI.6               Channel       ==N1.DI1.6
      AC450B.DI1-DI.7               Channel       ==N1.DI1.7
```

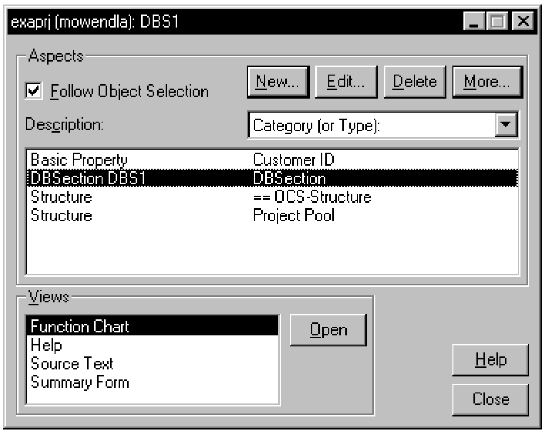
	Object	Description	Type	Created	Created By	Modifie
1	AC450B.DI1-DI.7	Basic Property	Basic Property	3.11.98 11:01	mowendla	3.11.98 11:
2	AC450B.DI1-DI.7	Structure	Structure	3.11.98 11:01	mowendla	03.11.98 1
3	AC450B.DI1-DI.7	Structure	Structure	3.11.98 11:01	mowendla	

Figure 4-47. Signals, Boards, and a Group of Boards in OCS Structure

After you have planned the I/Os the absolute reference designation of the boards and signals in the control system structure is defined and all the board and signal DB elements of a node or a part of a node are represented accumulated by an aspect of type DB Section.

An Advant object representing the node or part of the node regarding the board and signal DB elements has the following principle structure.

One aspect of type DB Section is supported for one object.



exaprj (mowendla): DBS1

Aspects

☑ Follow Object Selection

Description: Category (or Type):

Basic Property	Customer ID
DBSection DBS1	DBSection
Structure	== OCS-Structure
Structure	Project Pool

Views

Function Chart
Help
Source Text
Summary Form

Figure 4-48. Object Window of Advant Object with DB Section Aspect

Limitations

Boards as described above can be kept in the library and can be instantiated manually. Latest before generating the DB Section all "create only" terminals have to be defined by editing these terminals with Function Chart Builder's DB editor or within source code editor.

The board type and the call name has to be fixed from the beginning on.

Take care not to include a PC section into type circuit of an aspect of type DB Element.

DB Element type circuits are visible for a user working in Function Chart Builder's circuit mode or node mode. You have to take care not to insert these type circuits using the **Edit | Create** command of Function Chart Builder. They are reserved for insertion into a circuit or node by the **Generate DB** command.

When you are working on a DB Element type circuit (for example, Function Chart Builder type circuit mode is started) this type circuit is protected for access by another user (for example, starting up Function Chart Builder type circuit mode on this type circuit too) by a lock message (mechanism known from Application Builder).

You have to take care not to break this lock message.

4.3.28.5 Generate DB

The generation and update of the board and signal data is done with the **Generate DB** command on the DB Section aspect of an Advant object representing all or a certain amount of the boards and signals of an OCS node.

The **Generate DB** command is registered by the OCS Control Configuration Functions for the aspect type DB Section.

Make sure that the following prerequisites are fulfilled before executing the **Generate DB** command on DB Section aspect, for example, of a node object in OCS structure:

- I/O planning in control system oriented structure has been done.
- An aspect of type DB Section has been created for an object representing all or a certain amount of the boards and signals of the node.

- The relative reference designations for the OCS Structure aspect of all objects including the node object are set.
  - The "==" prefix is set in the OCS Structure aspect for all objects.
  - The '<' control for the relative reference designation is set in the OCS Structure aspect and for all channel objects.
  - The relative reference designation entered into the OCS Structure aspect of the board, channel, and other DB element related objects result in valid item designations for the DB elements.

  Note that the relative reference designation entries used to construct the item designation must not contain dots. The separating dots are supplied by the generation mechanism for the absolute reference designation in Structure Builder.

  For boards copied from example library the channels already contain the right entries.
- Parameter values have been edited in / imported into the parameter aspects.

You can activate the **Generate DB** command from the command popup menu of the **Object Window** of an Advant object having an aspect of type DB Section.

The **Generate DB** command instantiates all DB type circuits (for boards, signal and other DB elements) represented by appropriate objects in the control system oriented structure beneath the Advant object carrying the DB Section aspect.

If the Function Chart view is used, the DB type circuits are instantiated into the circuit of the DB Section aspect. Instantiation is done with default values on the next free location. After instantiation of a type circuit the symbol values for the type circuit instance are updated.

The update function takes care of all parameter references of the DB type circuits instantiated within the circuit. The circuit can be inserted/located into node of Function Chart Builder using drag and drop with the **Ctrl** key pressed (see Section 4.3.24, Drag and Drop (Insert)).

If the Node Function Chart view is used, the DB type circuits are instantiated into the node of the Node object above the object carrying the DB Section aspect. Instantiation is done with updated symbol values on the next free location.

The update function takes care of all parameter references of the DB type circuits instantiated via the DB section.

The following changes are taken care of by using the **Generate DB** command again after the changes have been introduced:

- New instance specific type circuits
- Additional board and signal objects.

Using the **Generate DB** with the Function Chart view again will completely skip the existing contents of the DB section circuit and insert the DB-TC instances of the DB section again.

Using **Generate DB** with the Node Function Chart view again will completely remove the DB-TC instances of the DB section from the node and insert the DB-TC instances of the DB section again.

DB elements created by type circuits are visible in node mode of Function Chart Builder.

Generate DB includes consolidation of virtual connections.

4.3.28.6 Update Parameter

See Section 4.3.27.2, Update Parameter.

4.3.28.7 Initialize

On DB Section Aspect

The **Initialize** command on a DB Section aspect (view Node Function Chart) deletes the instantiated DB element TC instances in the node of the Node Control Software aspect of the node object in the OCS Structure which either holds the DB Section aspect or is ancestor of the object holding the DB Section aspect.

On DB Element Aspect

The **Initialize** command on a DB Element aspect (view Node Function Chart) initializes a set of instantiated DB element TCs in the node object the current object had been instantiated into. This set of instantiated TCs is either defined by the objects in the OCS substructure on the current level and below having a DB Element aspect or if there is no OCS structure placement by the DB element aspect of the current object itself. The DB element TC(s) had been instantiated, for example by the **Generate DB** command executed with view Function Chart on this DB Element aspect. For each member of this set of instantiated DB element TCs the command executes the following in the Node Control Software aspect of the corresponding node object:

1. Updates parameters with a parameter reference to Basic Property name of category Customer ID to a blank name (if there is only one member in the set and the member represents a signal).
2. Deletes the instantiated DB element TC.

4.3.29 Execution of Commands by the Function Chart Builder

The Function Chart Builder executes such commands as **Generate DB**, **Update Parameter**, and **Initialize** in cooperating mode.

During their execution the Function Chart Builder can stay visible on the screen if it is started on the node or circuit or type circuit that is the destination for these commands.

While the command is executed, the Function Chart Builder's menus cannot be used, but the Function Chart Builder's windows are updated with the command results.

If you use a Function Chart Builder with a version lower or equal to 4.5/2 and this Function Chart Builder is already started for another node or circuit or type circuit, you get an error message and the command is not performed. Function Chart Builder of these versions can only be started once on a computer.

4.3.30 Export Process

The characteristic of the export process is to extract bulk data from AEW projects to a data storage outside AEW. This process is described in the *Bulk Data Handling and Reuse Reference Manual*.

The data scope is Advant Objects and their related aspects. The export range can be from one single object up to all objects of one project. Exported data follow the format of the Import function and can (re-)imported into any AEW project.

During the export process an auxiliary Application Builder export project named EOAPB has to be created on the selected export directory by acknowledging the **New Project** dialog of Application Builder (Figure 4-6) and by acknowledging a following query message box to create the corresponding project root.

To this export project EOAPB the functions supporting the export of ACC aspects write the aspect data.

4.3.31 Import Process

The characteristic of the import process is to load bulk data into an AEW project. This process is described in the *Bulk Data Handling and Reuse Reference Manual*.

4.3.31.1 Create Objects and Aspects

When you use the Structure Builder's **Project | Import** menu command, one of the given possibilities is to create new objects and aspects. The objects and aspects are stored in Import Data Files, flat ASCII files with *.csv format. In these files, one row describes the data for one object or aspect.

The meaning of the attributes for the ACC related aspects is described in this section

- **Create Node Control Software aspects**
  Adds a Node Control Software aspect to an existing Advant object.

  Attributes **EOName** and **EONameCategory** identify the Advant object.
  Attribute **AspectSystem** contain the value `EOAPB`.
  Attribute **AspectType** contains the value `Node Control Software`.
  Attribute **AspectDescription** contains the aspect description (can be empty).

  Additional aspect type related attributes are:

Name	mandatory parameter, file name identifying aspect data
NetNode **BusStation** **RNetRNode** **Posiion** **FCBVersion**	mandatory parameter only for node types of MP 200 or AC 400
NodeType	mandatory parameter

Comment
Option1 … **Option15**

- **Create Circuit, Type Circuit, DBSection or DBElement aspects**
  Adds a Circuit, Type Circuit, DBSection or DBElement aspect to an existing Advant object.

  Attributes **EOName** and **EONameCategory** identify the Advant object.
  Attribute **AspectSystem** contain the value `EOAPB`.
  Attribute **AspectType** contains the aspect type i.e. `Type Circuit`.
  Attribute **AspectDescription** contains the aspect description (can be empty).

  Additional aspect type related attributes are:

Name	mandatory parameter, file name identifying the aspect data
FCBVersion	
NodeType	mandatory parameter
Comment	
Option1 … **Option15**	

The values of these parameters are strings exactly those entered interactively in the Application Builder's dialogs.

Example for NodeType: `AC 410 with SW QC01-BAS11* 9.0`

NOTE

There is a blank between the node type name (`AC 410 with SW QC01-BAS11*`) and node type version (`9.0`). The same rule applies for the node type options described via the Option*n* parameter where *n*=1, …, 15.

Example for Function Chart Builder version: `4.5/1`

In addition, some of the AMPL Control Configuration aspects store data in files, such as source code files (*.cis, *.aax, etc.). If these files are provided during import, they are also copied to the destination project. The source file location must be specified in the import data file in the Data4 field by the full path to the file(s), for example c:\srcfiles.

The value of the Name attribute is used to locate the files on this directory:

- For a node the files are expected on

  <Data4>\<Name>\PCDATA regarding .AAX and .TIX files and
  <Data4>\<Name>\DBDATA regarding .BAX and .TBX and .CFX files

- For type circuits or circuits the files are expected on

  <Data4>\<Name> regarding .TCS or .CIS files

Figure 4-49 shows an example of an import data file.

```
EOName,EONameCategory,EOType,AspectSystem,AspectType,Category,AspectDescription,
Data1,Data2,Data3,Data4,Data5,Name,FCBVersion,NodeType,Comment,Option1,Option2,
Option3,Option4,Option5,Option6,Option7,Option8,Option9,Option10,Option11,Option12,
Option13,Option14,Option15

AC70,Customer ID,Node,EOAPB,DBElement,,,,,,c:\srcfiles,,AC70,4.5/1,AC 70 1.1,,OPT1:
Events 1.0,OPT2: extended PC 1.1,OPT3: Modbus I RTU 1.0,,,,,,,,,,,,

AC110,Customer ID,Node,EOAPB,DBElement,,,,,,c:\srcfiles,,AC110,4.5/1,AC 110
2.2,,OPT1: extended PC 2.2,OPT2: MVI Handling 2.2,OPT3: Events 2.2,OPT4: Modbus I RTU
2.2,OPT5: RCOM Driver 2.2,,,,,,,,,,
```

Figure 4-49. Example Import Data File for Create a Circuit Aspect

4.3.31.2 Create Objects and Aspects From Exported Data

A similar functionality as described in Section 4.3.31.1, Create Objects and Aspects is supported for exported Advant objects with ACC related aspects. In this case the Import Data Files have been written by the export process. During import the exported objects with the aspects are created in the current project and the aspect data of the ACC related aspects are copied from the export project named EOAPB to the current project.

The usage information for circuits inserted into nodes in FCB is kept. Depending on the context in the current project this might not be valid. It can be corrected using the Initialize command.

Type circuit references are handled by copying the reference. Depending on the available project specific type circuits this may lead to the attempt to use the project specific type circuit of the same name in Generate DB, Update Parameter, Locate Circuit or drag and drop.

4.4 Runtime Operation Menus

This section describes the commands (buttons, menu items of the command popup menu) available in the **Object Window** for OCS Control Configuration aspect types.

4.4.1 Command Buttons

An example **Object Window** is shown in Figure 4-2.

4.4.1.1 New…

Use this command button to add a new aspect to an Advant object:

The **New Aspect** dialog is displayed, see Figure 4-4.

You can add a Project OCS Defaults aspect to an object representing a project, a Node Control Software aspect to an object representing a node and a Type Circuit or Circuit aspect to an object representing a physical object.

4.4.1.2 Close

Use this command to close the **Object Window**.

4.4.1.3 Delete

Use this command button to delete an existing aspect of an Advant object.

Messages are shown as in Figure 4-34 and Figure 4-24.

You have to delete all Type Circuit, Circuit and Node Control Software aspects of a project before you are allowed to delete the Project OCS Defaults aspect.

4.4.1.4 Edit…

Use this command button for changes of the aspect description of an existing aspect.

4.4.1.5 More…

Use this command button to display the command popup menu.

Mouse short cut for this command is clicking the right mouse button on the aspect in the aspect list of the **Object Window**.

4.4.1.6 Open

Use this command button to open an aspect with a selected view.

Mouse short cut for this command is double clicking the left mouse button on the aspect in the aspect list of the **Object Window**.

See Table 4-5 for views of aspect types to be opened by this command.

4.4.2 Popup Menu Commands

An example **Object Window** with displayed popup menu command is shown in Figure 4-3. For each command an additional **Help** command is available. When you click **Help**, the corresponding help topic will be displayed.

4.4.2.1 Attach

Use this command to attach existing aspect data (by copying) to an aspect of type Project OCS Defaults, Type Circuit, Circuit, DB Element, DB Section, or Node Control Software.

The command provides the **File | Copy** function of Application Builder.

The existing data to be attached have to be available in known ABB Master standard directories.

4.4.2.2 Convert

Use this command to convert source code data of a Node Control Software aspect.

The command provides the **File | Convert** function of Application Builder.

The Application Builder converts source code files generated by Function Chart Builder (general representation; 8-bit ANSI or ISO88591) to ABB Master representation (7-bit character).

4.4.2.3 Extract

Use this command to extract source code data of a Node Control Software aspect.

The command provides the **File | Extract** function of Application Builder.

Than Application Builder extracts a specified section of the source code file generated by Function Chart Builder (general representation; 8-bit ANSI or ISO88591) and converts it to ABB Master representation (7-bit character).

4.4.2.4 Generate DB

Use this command to accumulate the data of the DB element aspects of board and signal objects in the OCS structure below the object carrying the DB Section aspect.

The **Generate DB** command instantiates all DB type circuits (for boards, signal and other DB elements) represented by appropriate objects in the control system oriented structure beneath the Advant object carrying the DB Section aspect into the Node Control Software aspect (view Node Function Chart) or into the DB Section aspect (view Function Chart).

Depending on the view the **Generate DB** command skips or deletes the existing contents of the DB section circuit and inserts then the DB-TC instances.

Generate DB includes consolidation of virtual connections.

4.4.2.5 Open

Use this command to open an aspect with a selected view.

Mouse short cut for this command is double clicking the left mouse button on the aspect in the aspect list of the **Object Window**.

Aspect Type	View
Project OCS Defaults	Summary Form
	Notes Text
	Circuit Usage List
	Help
Type Circuit	Summary Form
	Function Chart
	Source Text
	Help
DB Element	Summary Form
	Function Chart
	Source Text
	Help
Circuit	Summary Form
	Function Chart
	Source Text
	Node Function Chart
	Help
DB Section	Summary Form
	Function Chart
	Source Text
	Help
Node Control Software	Summary Form
	Node Function Chart
	Source Text
	On-line
	Communication Configuration
	Help

Table 4-5. Open Command: Aspect Types and Views

4.4.2.6 Print

Use this command to print aspect data with a selected view. The command provides the **File | Print** function of Application Builder and Function Chart Builder. See views that can be printed in the following table:

Aspect Type	Views
Project OCS Defaults	Node List, Circuit List, Type Circuit List
Type Circuit	Documents
DB Element	Documents
Circuit	Documents
DB Section	Documents
Node Control Software	Node Documents

Table 4-6. Print Command: Aspect Types and Views

4.4.2.7 Set Options

Use this command to set options for your project work with the AMPL Control Configuration functions. For each Project OCS Defaults aspect you can define by the options the following:

- The way names of objects are proposed by Application Builder
- Immediate **Copy** operation with proposed names without dialog upon **Attach** command
- Deletion of type circuit, circuit, and node data or just deletion of the assignment of data to the corresponding aspect upon **Delete** command
- Restrict copying to the source code, for example, upon **Attach** command.

4.4.2.8 Split

Use this command to split source code data of a Node Control Software aspect. The command provides the **File | Split Source Code** function of Application Builder. The Application Builder splits large DB or PC source code files to continuation files for download by the On-line Builder.

4.4.2.9 Update Parameter

Use this command to update the parameters of a Circuit aspect or all Circuit aspects in the project or the parameters of all DB Element aspects belonging to a DB Section aspect.

The command provides the **Edit | Symbol Values** function of Function Chart Builder and the consolidation of virtual connections.

The parameter reference update refers to providing the actual parameter value as the symbol value for the symbol that has the companion reference symbol which specifies the parameter reference within the type circuit instance.

4.4.2.10 Initialize

Use this command to initialize aspects of an object.

In case of a DB Section aspect the Function Chart Builder deletes instantiated DB element TC instances in the node of the Node Control Software aspect of the node object in the OCS Structure which either holds the DB Section aspect or is ancestor of the object holding the DB Section aspect.

In case of a Circuit aspect the Function Chart Builder deletes the circuit contents in the node of the Node Control Software aspect of the node object.

In case of a Node Control Software aspect, the command is executed on all Circuit aspects of objects placed in the OCS structure below the object have the Node Control Software aspect.

4.4.2.11 Concatenate

Use this command to concatenate data of a Node Control Software aspect.

The command provides the **File | Concatenate** function of Application Builder.

The Application Builder concatenates continuation DB or PC source code files delivered by the On-line Builder to one single file.

4.4.2.12 Lock/Unlock

Use these commands to lock/unlock explicitly Node Control Software, Circuit, or Type Circuit, DB Element, or DB Section aspects of an object.

By locking you prevent these aspects explicitly from access by other users.

The commands provide the **File | Lock/Unlock** function of Application Builder.

NOTE

While you are working on a Node Control Software, Circuit, or Type Circuit aspect with some AMPL Control Configuration component, the object is locked automatically for you. If you stop working with the component, the object will also be unlocked automatically.

4.4.2.13 Locate Circuit

Use this command to insert a circuit into the PC program of a node via the reference designation.

The command is available for Circuit and Node Control Software aspects.

When you use the command on a Circuit aspect, you insert the contents of the circuit file into the PC program of the node, specified by the relative reference designation of the OCS structure aspect of the object that contains the Circuit aspect.

Use the command on a Node Control Software aspect to locate all Circuit aspects of objects placed in the OCS structure below the object having the Node Control Software aspects.

4.4.2.14 Update FCB Data

Use this command to upgrade or change the Function Chart Builder version for all AMPL Control Configuration aspects of the project.

For each of these aspects the command automatically performs the steps needed to change Function Chart Builder version:

1. Start FCB of current (older) version
2. Generate source code
3. Exit FCB
4. Modify the tool version
5. Start FCB of the new version
6. Initialize internal data storage (ODB) of FCB and backtranslate source code
7. Exit FCB

Prerequisite for using this command is that both versions of the FCB are installed.

Chapter 5 Maintenance

5.1 Error Messages

During the use of the AMPL Control Configuration functions, the Structure Builder presents its own system messages and the system messages of Application Builder, Function Chart Builder, and On-line Builder pertaining to the integrated functionality.

The AMPL Control Configuration functions show two kinds of error messages:

- Error messages of components of the AMPL Control Configuration product, that is, system messages of Application Builder, Function Chart Builder, and On-line Builder pertaining to the integrated functionality.

  The error messages appear in dialog boxes containing

  - The title stating the tool name, for example, `Function Chart Builder`
  - An error message identifier, for example, `#12291`
  - An icon reflecting the severity of the message, for example, a **STOP** sign
  - Concise statement of the problem
  - A hint to circumvent the problem
  - **OK** button to close the dialog box.

  Follow the hint.

- Error messages of the interface between Structure Builder and components of the AMPL Control Configuration product.

  The error messages appear in dialog boxes containing

  - The title stating the interface name, for example, `Structure Builder - CCS`
  - Eventually an icon reflecting the severity of the message, for example, a **STOP** sign
  - Concise statement of the problem
  - Eventually a hint to circumvent the problem
  - **OK** button to close the dialog box

  Follow the hints given in the message.

5.2 Log-File for Update Parameter/Generate DB/Initialize

For tracing parameter settings or element deletions in **Update Parameter**, **Generate DB**, **Initialize** commands, and in the Update Parameter sequence included into drop object with circuit aspect you can specify writing a log-file. Edit the `APB.INI` file in the `ADVABLDENV` directory:

```
...
[SetApbMessage]
PrintLogFile=n
```

n=`0` stands for switching off generation of the log-file,
n=`1` stands for deleting the old log-file and switching on the generation of a new log-file,
n=`2` stands for generation of the new log-file and appending it to the old one.
`BatchMode=b`

b=`0` stands for interactive mode,
b=`1` stands for batch mode where the interactive error messages are suppressed except for start and end messages.

In case of the Update Parameter or Generate DB commands this log-file lists the results of parameter reference executions for the aspect instances, in case of the Initialize command it lists the initialized aspect instances.

The log-file is named `PARFCB.LOG` and stored in the `ADVABLDENV` directory.

Starting with ACC 1.7

- The log-file is stored in the TEMP directory.
- By default the settings PrintLogFile=2 and BatchMode=1 are in effect (no editing APB.INI needed).
- Messages that refer to the log-file allow to start up AS100EDIT on the log-file directly.

Header Line

The log-file starts with a header line stating the start date and time and ends with an end line stating the end date and time.

Example:

```
Logfile of  Application Builder / EOAPB, Start Time: Fri Dec 11 12:48:50 1998
...
Fri Dec 11 13:03:53 1998
```

Following the header line a section lists

- Project name
- Object name
- Aspect type
- Command
- View
- Target object name
- Target aspect type.

Example

```
Project Name = deltest1
    Object Name = CBAI
        Aspect Type = DBE.Aspect
    Command = Update Parameter
        View = NODE_FUNCTION_CHART
    Target Object Name = NewNode2
        Target Aspect Type = Node Control Software
```

The object name identifies the object that carries the aspect on which the command has been launched (for example the object carrying the Circuit aspect on which the **Update Parameter** command has been executed).

The target object name identifies the object that carries the aspect against which the command works (for example the object carrying the Node Control Software aspect against which the **Update Parameter** command executed on the Circuit aspect works).

Initialize Command Results

In case of the **Initialize** command a list of aspect instances (listed by instance name), that have been removed from Function Chart Builder, is printed. Success is shown by +OK, failure is shown by -OK.

Example:

```
>>> Removing DB Type Circuit instances
CBAI_DBElement       +OK
CSAI1_DBElement      +OK
CSAI10_DBElement     +OK
CSAI11_DBElement     +OK
CSAI12_DBElement     +OK
CSAI13_DBElement     +OK
CSAI14_DBElement     +OK
CSAI15_DBElement     +OK
CSAI16_DBElement     +OK
CSAI2_DBElement      +OK
CSAI3_DBElement      +OK
CSAI4_DBElement      +OK
CSAI5_DBElement      +OK
CSAI6_DBElement      +OK
CSAI7_DBElement      +OK
```

```
CSAI8_DBElement      +OK
CSAI9_DBElement      +OK
<<< Removing DB Type Circuit instances
Initialize finished without errors
```

Update Parameter and Generate DB Command Results

In case of **Update Parameter** and **Generate DB,** the result section begins with two lines stating type circuit name and instance name and then lists the results of parameter reference execution against the data source (Parameter Builder or Structure Builder) and against the data sink (Function Chart Builder) in a table.

The table's columns are the following:

- The parameter's name in the parameter aspect (PARName),
- The result status of the retrieval (Ret),
- The value of the parameter (PARvalue),
- The name of the TC symbol that receives the value, and
- The result status of the single edit symbol value action (EditSymValue).

The result status is either +OK for success or -OK for failure.

Example for the **Generate DB** command:

```
>>> InstantiateTCNFC

Type Circuit Name = CBAI
Instance Name = CBAI_DBElement



PARName       Ret      PARValue   SymbolName EditSymValue
-------------+-------+----------+--------------+-----------------
$ABS          +OK      AI5        binstance     +OK
CONV_PAR      +OK      0..10V     CONV_PAR      +OK
IMPL          +OK      0          IMPL          +OK

TC No = 1   is stored  +OK
                       ...
```

```
Type Circuit Name = CSAI9
Instance Name = CSAI9_DBElement

PARName      Ret     PARValue   SymbolName EditSymValue
-------------+--------+--------------+-----------+------------
ACT          +OK     1          ACT        +OK
CONV_PAR     +OK     0..10V     CONV_PAR   +OK
$DESCRIPTION +OK     Basic PropertyDESCR   +OK
RANGEMAX     +OK     100.0      RANGEMAX   +OK
RANGEMIN     +OK     -100.0     RANGEMIN   +OK
$NAME        +OK     CSAI9      sname      +OK
$ABS         +OK     AI5.9      sinstance  +OK
UNIT         +OK     V          UNIT       +OK
TC No = 17   is stored  +OK
SaveODB for DB Section is OK
<<< InstantiateTCNFC
<<< APBIGenerateDB
Generate DB performed successfully.
```

Example for the **Update Parameter** command:

In case of a problem with execution of a parameter reference the parameter reference string is printed in the listing line before the line describing data source and sink.

Example: taken from Update Parameter in Drag & Drop / Locate Circuit

```
PARName      Ret     PARValue   SymbolName EditSymValue
-------------+--------+--------------+-------------+------------
::$$REF.ProcessObjectParameter.FB1_USED_EXT::
FB1_USED_EXT  +OK                      FB1_USED_EXT-OK
::$$REF.ProcessObjectParameter.FB2_USED_EXT::
FB2_USED_EXT  +OK                      FB2_USED_EXT-OK
::$$REF.ProcessObjectParameter.GROUP::
GROUP         +OK                      GROUP       -OK
...
```

The final check result for storing the TC with changed parameters is listed in the line after the result section by stating a TC number and the check status.

The check status is either +OK for success or -OK for failure. This check corresponds to pressing the OK button in Function Chart Builder's Edit Symbol Value dialog.

NOTE

Parameter names starting with $ such as $DESCRIPTION, $NAME and $ABS are used for parameters that are stored in Structure Builder, $ABS describes that a special algorithm is used to derive the parameter value from the absolute reference designation of the object within the OCS structure.

In some cases the commands Start dialog is also printed at the start of the log-file. In cases where commands give a sum up problem message, it is also printed at the end of the results section.

5.3 Important Environment Variables

5.3.1 Environment Variable TC$LEV1

For using type circuits of a library project the environment variable TC$LEV1 has to be set. It is recommended to set it to the project specific TCDATA directory of the library project, and not to some separate TCLEV1 directory. This does not require moving or copying TCS files to this directory.

Example

```
TC$LEV1=L:\DATA\EX_LIB1\NODES\PR_TC\TCDATA
```

NOTE

TC$LEV1 must be a system environment variable and the definition should be written in upper case letters. For using type circuits of a second library project the environment variable TC$LEV2 has to be set accordingly.

5.3.2 Environment Variable ADVABLDENV

If used, the ADVABLDENV environment variable must be a system environment variable. It is set by the setup program.

5.3.3 Environment Variables TMP and TEMP

The environment variables TMP and TEMP should be system environment variables.

Otherwise Function Chart Builder functions accessed via OLE write their temporary files to the WINNT directory.

Appendix A References

A.1 Aspect Types and Integrated Functions

Table A-1 summarizes the aspect types, the integrated AMPL Control Configuration functions, and the views. Every command has an additional **Help** view, which is not listed in the table.

Table A-1. Summary of Aspect Types, Integrated Functions, and Views

Advant Object Representing	Aspect Types	Integrated Functions (Commands)	Views
Project	Project OCS Defaults	New, Delete	Summary Form
		More:	
		Open	Circuit Usage List, Notes Text, Summary Form
		Print	Node List, Circuit List, Type Circuit List
		Attach	Defaults
		Set Options	Defaults
		Update Parameter	Function Chart
		Update FCB Data	Summary Form
Physical object	Type Circuit	New, Delete	Summary Form
		More:	
		Open	Function Chart, Source Text, Summary Form
		Print	Documents
		Attach	Function Chart
		Lock/Unlock	Summary Form

Table A-1. Summary of Aspect Types, Integrated Functions, and Views

Advant Object Representing	**Aspect Types**	**Integrated Functions (Commands)**	**Views**
Board or signal or other kind of DB element	DB Element	New, Delete	Summary Form
		More:	
		Open	Function Chart Source Text Summary Form
		Print	Documents
		Attach	Function Chart
		Update Parameter	Node Function Chart
		Generate DB	Node Function Chart
		Initialize	Node Function Chart
		Lock/Unlock	Summary Form
Physical object	Circuit	New, Delete	Summary Form
		More:	
		Open	Function Chart Node Function Chart Source Text Summary Form
		Print	Documents
		Attach	Function Chart
		Update Parameter	Function Chart Node Function Chart
		Locate Circuit	Node Function Chart
		Initialize	Node Function Chart
		Lock/Unlock	Summary Form
		Insert (by drag and drop)	Node Function Chart

Table A-1. Summary of Aspect Types, Integrated Functions, and Views

Advant Object Representing	Aspect Types	Integrated Functions (Commands)	Views
Node or part of node	DB Section	New, Delete	Summary Form
		More:	
		Open	Function Chart Source Text Summary Form
		Print	Documents
		Attach	Function Chart
		Update Parameter	Function Chart Node Function Chart
		Generate DB	Function Chart Node Function Chart
		Initialize	Node Function Chart
		Lock/Unlock	Summary Form
Node	Node Control Software	New, Delete	Summary Form
		More:	
		Open	Communication Configuration Node Function Chart Online Source Text Summary Form
		Print	Documents
		Attach	Node Function Chart
		Update Parameter	Node Function Chart
		Locate Circuit	Node Function Chart
		Convert, Extract, Split, Concatenate	Node Function Chart
		Lock/Unlock	Summary Form
		Initialize	Node Function Chart

A.2 Commands and Views of Aspect Types

These tables describe all the available commands and the available views for each aspect type, and comment them.

Table A-2. Project OCS Defaults Aspect Type, Commands and Views

Command (Button)	Command (Popup Menu Item)	View	Remark: Function Corresponds to
New…		[Summary Form]	APB File | New | Project including “Set Defaults for the Project” and “More Settings.”
Delete		[Summary Form]	APB File | Delete Project
Open		Summary Form Notes Text Circuit Usage List	APB Option | Project Defaults APB File | Open | Project Notes Text Editor opening file CiStatus.txt
More…	Print	Node List Circuit List Type Circuit List	APB File | Print with view Node List APB File | Print with view Circuit List APB File | Print with view Type Circuit List
	Attach	[Summary Form]	APB File | Copy (scope Project) restricted to Project OCS Defaults contents (Project Defaults and Project Notes)
	Set Options	[Options]	Set Options dialog
	Update Parameter	Function Chart	Update all parameters of all instantiated TCs in all circuits within the project scope.
	Update FCB Data	[Summary Form]	APB File | Update FCB Data | Project

Table A-3. Type Circuit Aspect Type, Commands and Views

Command (Button)	**Command (Popup Menu Item)**	**View**	**Remark: Function Corresponds to**
Add…		[Summary Form]	APB File | New | Circuit
Delete		[Summary Form]	APB File | Delete Selected
Open		Function Chart	APB Tools | Function Chart Builder, detail view selection within FCB: • PC Section (Function Chart <-> Tree) • DB Section • Symbol (Parameter) • TC/CI-Terminals.
		Summary Form	APB File | Modify TC
		Source Text	Text Editor
More…	Print	[Documents]	FCB File | Print
	Attach	[Function Chart]	APB File | Copy (scope Type Circuit)
	Lock/Unlock	[Summary Form]	APB File | Lock/Unlock

Table A-4. DB Element Aspect Type, Commands and Views

Command (Button)	**Command (Popup Menu Item)**	**View**	**Remark: Function Corresponds to**
Add…		[Summary Form]	APB File | New | Type Circuit
Delete		[Summary Form]	APB File | Delete Selected
Open		Function Chart Summary Form Source Text	APB Tools | Function Chart Builder, detail view selection within FCB: • PC Section (Function Chart <-> Tree) • DB Section • Symbol (Parameter) • TC/CI-Terminals. APB File | Modify TC Text Editor
More…	Print	[Documents]	FCB File | Print
	Attach	[Function Chart]	APB File | Copy (scope Type Circuit)
	Update Parameter	Node Function Chart	Update the parameters of all DB-TCs of the OCS structure below and including the Advant object having the DB Element aspect in the node’s DB section.
	Generate DB	Node Function Chart	Instantiate all DB-TCs of the OCS structure below and including the Advant object having the DB Element aspect into the node’s DB section, and update the parameters.
	Initialize	Node Function Chart	Update parameters with Basic Property name references to empty name (id single signal) and delete the instantiated DB-TCs in the circuit of the corresponding DB section.
	Lock/Unlock	[Summary Form]	APB File | Lock/Unlock

Table A-5. Circuit Aspect Type, Commands and Views

Command (Button)	**Command (Popup Menu Item)**	**View**	**Remark: Function Corresponds to**
Add…		[Summary Form]	APB File | New | Circuit
Delete		[Summary Form]	APB File | Delete Selected
Open		Function Chart	APB Tools | Function Chart Builder, detail view selection within FCB: • PC Section (Function Chart <-> Tree) • DB Section • Symbol (Parameter) • TC/CI-Terminals.
		Summary Form	APB File | Modify CI
		Source Text	Text Editor
		Node Function Chart	APB Tools | Function Chart Builder, additionally detail view and selection are set for FCB to Function Chart and the FUNCM of the circuit in the PC program.
More…	Print	[Documents]	FCB File | Print dialog
	Attach	[Function Chart]	APB File | Copy (scope Circuit)
	Update Parameter	Function Chart	Update the parameters of the TCs in the circuit.
		Node Function Chart	Update the parameters of the circuit inserted in the node: • Instance name of the FUNCM representing the circuit in the node • Parameters of the TCs instantiated in the circuit.
	Initialize	[Summary Form]	Delete circuit from PC program of a node (if used), Circuit aspect not touched
	Lock/Unlock	[Summary Form]	APB File | Lock/Unlock
	Locate Circuit	Node Function Chart	Insert (locate and expand) circuit into PC program of a node. Location is defined via reference designation.
Drag and drop (mouse action)	(Insert)	Node Function Chart	Insert (locate and expand) circuit into PC program of a node

Table A-6. DB Section Aspect Type, Commands and Views

Command (Button)	**Command (Popup Menu Item)**	**View**	**Remark: Function Corresponds to**
Add…		[Summary Form]	APB File | New | Circuit
Delete		[Summary Form]	Delete DB-TCs from node (if instantiated) and delete DB Section aspect; use also APB File | Delete Selected circuit).
Open		Function Chart	APB Tools | Function Chart Builder, detail view selection within FCB: • PC Section (Function Chart <-> Tree) • DB Section • Symbol (Parameter) • TC/CI-Terminals.
		Summary Form	APB File | Modify CI
		Source Text	Text Editor
More…	Print	[Documents]	FCB File | Print dialog
	Attach	[Function Chart]	APB File | Copy (scope Circuit)
	Update Parameter	Function Chart	Update the parameters of all board and signal TCs of the OCS structure below the Advant object having the DB Section aspect into the circuit of the DB Section aspect.
		Node Function Chart	Update the parameters of all DB- TCs of the OCS structure below and including the Advant object having the DB Section aspect in the corresponding node.
	Generate DB	Function Chart	Instantiate all DB-TCs of the OCS structure below the Advant object having the DB Section aspect into the DB section circuit and update the parameters.
		Node Function Chart	Instantiate all DB-TCs of the OCS structure below and including the Advant object having the DB Section aspect into the corresponding node and update the parameters.
	Initialize	[Summary Form]	Delete DB-TCs from node (if instantiated), DB Section aspect not touched.
	Lock/Unlock	[Summary Form]	APB File | Lock/Unlock

Table A-7. Node Control Software Aspect Type, Commands and Views

Command (Button)	**Command (Popup Menu Item)**	**View**	**Remark: Function Corresponds to**
Add…		[Summary Form]	APB File | New | Node
Delete		[Summary Form]	APB File | Delete Selected (scope Node)
Open		Node Function Chart	APB Tools | Function Chart Builder Detail view selection within FCB: • PC Section (function chart<->tree • DB Section
		Summary Form	APB File | Modify | Node
		On-line	APB Tools | On-line Builder (ONB for AC400 controllers and MP200 process stations)
		Source Text	APB Extended View dialog, base to start Text Editor
More…	Print	Node Documents	FCB File | Print
		Circuit Usage List	Print list with contents of circuit usage administration for all Circuit aspects used in this node.
	Attach	[Summary Form]	APB File | Copy (scope Node)
	Update Parameter	Node Function Chart	Update the parameters of all circuits inserted in the node (node mode): • Instance names of the FUNCMs representing the circuits in the node • Parameters of the TCs instantiated in these circuits.

Table A-7. Node Control Software Aspect Type, Commands and Views

Command (Button)	**Command (Popup Menu Item)**	**View**	**Remark: Function Corresponds to**
More…	Convert	[Source Text]	APB File | Convert
	Extract	[Source Text]	APB File | Extract
	Split	[Source Text]	APB File | Split
	Concatenate	[Source Text]	APB File | Concatenate
	Lock/Unlock	[Summary Form]	APB File | Lock/Unlock
	Locate Circuit	Node Function Chart	Insert (locate and expand) the circuits of all objects placed in the OCS structure beneath the node object into PC programs of the node. Location is defined via reference designation.
	Initialize	Node Function Chart	Delete the circuits of all objects placed in the OCS structure beneath the node object from the PC programs of a node (if used), Circuit aspects not touched

INDEX

A

Advant Object Technology 3-1
AS100-Edit 4-12
Attach 4-15, 4-60
  Circuit 4-15
  DB Element 4-15
  DB Section 4-15
  Node Control Software 4-25
  Project OCS Defaults 4-11
  Type Circuit 4-15

B

Bold strings 1-4
Bulk Data 4-8

C

Circuit 1-9, 3-2, A-2
  Attach 4-15, 4-60
  Copy 4-16
  Delete 4-22, 4-60
  Drag and Drop 4-35
  Initialize 4-38
  Locate Circuit 4-39, 4-64
  Lock/Unlock 4-64
  New 4-59
  Open 4-18
  Print 4-21
  Update Parameter 4-38, 4-45, 4-63
Circuit Handling 3-4
Circuit Usage List 4-41
Close
  Object Window 4-60
Command syntax 1-4
Communication Configuration
  View 4-27
Concatenate 4-64
Conventions 1-4
Convert 4-5, 4-61
  Node Control Software 4-30
Convert Source Code Files 4-30 to 4-31
Cooperating 1-1
Copy 4-11, 4-15, 4-25
  Node Control Software 4-25
Copy Aspect 4-16
Copy Source Code Only 4-17, 4-25 to 4-26
Create
  Advant Objects 4-1

D

DB Element 1-9, 3-2, A-2
  Attach 4-15
  Copy 4-16
  Delete 4-22
  Generate DB 4-48
  Initialize 4-56
  Lock/Unlock 4-64
  New 4-13
  Open 4-18
  Print 4-21
  Update Parameter 4-47
DB Element Handling 3-3, 4-49
DB Section 1-9, 3-2, A-3
  Attach 4-15
  Copy 4-16
  Delete 4-22
  Generate DB 4-48
  Initialize 4-56
  Lock/Unlock 4-64
  New 4-13
  Open 4-18
  Print 4-21
  Update Parameter 4-47
Delete 4-60
  Node Control Software 4-29
Delete an Advant Object 4-3
Delete an Aspect 4-5, 4-22
Documentation 1-5
Drag and Drop 4-48
Duplicate 4-3

E

Edit 4-60
Enter key 1-4
Extended View 4-27
Extract 4-5, 4-61
  Node Control Software 4-30

G

Generate DB 4-47, 4-61

I
Import 4-57
Independent 1-1
Initialize 4-64
    Circuit 4-38
    DB Element 4-56
    DB Section 4-56
Installation 2-1
Italic strings 1-4

K
Keyboard interaction 1-4

L
Locate Circuit 4-39, 4-64
Lock 4-64
Lock/Unlock 4-32
Logfile 5-2

M
Modify 4-19
More 4-60

N
New 4-13 to 4-14, 4-59
    Circuit 4-14
    DB Element 4-13
    DB Section 4-13
    Node 4-23
    Type Circuit 4-14
New Aspect 4-13, 4-59
Node Control Software
    Attach 4-25, 4-60
    Available Functions 1-9
    Concatenate 4-64
    Convert 4-61
    Copy 4-25
    Definition of Aspect 3-2
    Delete 4-29, 4-60
    Extract 4-30, 4-61
    Locate Circuit 4-39, 4-64
    Lock/Unlock 4-64
    New 4-23, 4-59
    Open 4-26
    Print 4-28
    Split 4-31, 4-63
    Update Parameter 4-46

O
Object Window 4-2, 4-47
    Close 4-60
    More 4-60
    Open an aspect 4-60
OCS Structure 4-53
Open 4-60 to 4-61
    Node Control Software 4-26
    Project OCS Defaults 4-11
Open Aspect 4-5, 4-18
Options Set 4-63

P
Parameter
    References 3-2
    Update 4-63
PARFCB.LOG 5-2
Print 4-5, 4-21, 4-63
    Node Control Software 4-28
    Project OCS Defaults 4-63
Project Defaults 4-12
Project OCS Defaults 1-9, 3-2, A-1
    Attach 4-11, 4-60
    Circuit Usage List 4-41
    Delete 4-13, 4-60
    Generate DB 4-48
    New 4-59
    Open 4-11
    Print 4-63
    Set Options 4-63
    Update Parameter 4-47

S
Set Options 4-63
Split 4-5, 4-63
    Node Control Software 4-31
Split Source Code 4-31
Summary Form 4-12, 4-19, 4-27
Syntax 1-4

T
Type Circuit 1-9, 3-2, A-1
    Attach 4-15, 4-60
    Copy 4-16
    Delete 4-22, 4-60
    Lock/Unlock 4-64
    New 4-59
    Open 4-18
    Print 4-21

U

Unlock 4-64
Update Parameter 4-45, 4-63
    Circuit 4-38

V

View
    Communication Configuration 4-27
    Function Chart 4-18
    Node Documents 4-29
    Node Function Chart 4-19, 4-26
    Notes Text 4-12
    On-line 4-27
    Source Text 4-20, 4-27
    Summary Form 4-12, 4-19, 4-27

3BSE 013 931R301

November 1999